Nmap 参考指南

wizardforcel

Published
with GitBook

目錄

介绍	0
一、 描述	1
二、 译注	2
三、 选项概要	3
四、 目标说明	4
五、 主机发现	5
六、 端口扫描基础	6
七、 端口扫描技术	7
八、 端口说明和扫描顺序	8
九、 服务和版本探测	9
十、 操作系统探测	10
十一、 时间和性能	11
十二、 防火墙/IDS 躲避和哄骗	12
十三、 输出	13
十四、 其它选项	14
十五、 运行时的交互	15
十六、 实例	16
十七、 Bugs	17
十八、 作者	18
十九、 法律事项(版权、许可证、担保(缺)、出口限制)	19

Nmap 参考指南

Table of Contents

- 描述
- 译注
- 选项概要
- 目标说明
- 主机发现
- 端口扫描基础
- 端口扫描技术
- 端口说明和扫描顺序
- 服务和版本探测
- 操作系统探测
- 时间和性能
- 防火墙/IDS
- 躲避和哄骗
- 输出
- 其它选项
- 运行时的交互
- 实例
- Bugs
- 作者
- 法律事项（版权、许可证、担保、出口限制）
- Unofficial Translation Disclaimer

Name

nmap — 网络探测工具和安全/端口扫描器

Synopsis

```
nmap [ 扫描类型 ...] [ 选项 ] { 扫描目标说明 }
```

一、 描述

Nmap (“Network Mapper(网络映射器)”) 是一款开放源代码的 网络探测和安全审核的工具。它的设计目标是快速地扫描大型网络，当然用它扫描单个 主机也没有问题。Nmap 以新颖的方式 使用原始 IP 报文来发现网络上有哪些主机，那些 主机提供什么服务(应用程序名和版本)，那 些服务运行在什么操作系统(包括版本信息)， 它们使用什么类型的报文过滤器/防火墙，以及一堆其它功能。虽然 Nmap 通常用于安全审核， 许多系统管理员和网络管理员也用它来做一些 日常的工作，比如查看整个网络的信息， 管理服务升级计划，以及监视主机和服务的运行。

Nmap 输出的是扫描目标的列表，以及每个目标的补充信息，至于是哪些信息则依赖于所使用的 选项。“所感兴趣的端口表格”是其中的关键。那张表列出端口号，协议，服务名称和状态。 状态可能是 open(开放的)，filtered(被过滤的)， closed(关闭的)，或者 unfiltered(未被过滤的)。 Open(开放的)意味着目标机器上的应用程序正在该端口监听连接/报文。 filtered(被 过滤的) 意味着防火墙，过滤器或者其它网络障碍阻止了该端口被访问，Nmap 无法得知 它是 open(开放的) 还是 closed(关闭的)。 closed(关闭的) 端口没有应用程序在它上面监听，但 是他们随时可能开放。 当端口对 Nmap 的探测做出响应，但是 Nmap 无法确定它们是关闭还是开 放时，这些端口就被认为是 unfiltered(未被过滤的) 如果 Nmap 报告状态组合 open|filtered 和 closed|filtered 时，那说明 Nmap 无法确定该端口处于两个状态中的哪一个状态。 当要求进行版本探测时，端口表也可以包含软件的版本信息。当要求进行 IP 协议扫描时 (-sO)，Nmap 提供关于所支持的 IP 协议而不是正在监听的端口的信息。

除了所感兴趣的端口表，Nmap 还能提供关于目标机的进一步信息，包括反向域名，操作系统猜 测，设备类型，和 MAC 地址。

一个典型的 Nmap 扫描如 Example 1, “一个典型的 Nmap 扫描”所示。在这个例子中，唯一的选 项是-A， 用来进行操作系统及其版本的探测，-T4 可以加快执行速度，接着是两个目标主机名

Example 1. 一个典型的 Nmap 扫描

```
# nmap -A -T4 scanme.nmap.org playground

[Starting nmap ( http://www.insecure.org/nmap/](http://www.insecure.org/nmap/) )
Interesting ports on scanme.nmap.org (205.217.153.62):
(The 1663 ports scanned but not shown below are in state: filtered)
port STATE SERVICE VERSION
22/tcp open ssh OpenSSH 3.9p1 (protocol 1.99)
53/tcp open domain
70/tcp closed gopher
80/tcp open http Apache httpd 2.0.52 ((Fedora))
113/tcp closed auth
Device type: general purpose
Running: Linux 2.4.X|2.5.X|2.6.X
OS details: Linux 2.4.7 - 2.6.11，Linux 2.6.0 - 2.6.11
Uptime 33。908 days (since Thu Jul 21 03:38:03 2005)

Interesting ports on playground。nmap。或者 g (192.168.0.40):
(The 1659 ports scanned but not shown below are in state: closed)
port STATE SERVICE VERSION
135/tcp open msrpc Microsoft Windows RPC
139/tcp open netbios-ssn
389/tcp open ldap?
![image](Nmap中文官方文档/Image_003.png)
445/tcp open microsoft-ds Microsoft Windows XP microsoft-ds
1002/tcp open windows-icfw?
1025/tcp open msrpc Microsoft Windows RPC
1720/tcp open H.323/Q.931 CompTek AquaGateKeeper
5800/tcp open vnc-http RealVNC 4.0 (Resolution 400x250; VNC TCP port: 5900)
5900/tcp open vnc VNC (protocol 3.8)
MAC Address: 00:A0:CC:63:85:4B (Lite-on Communications)
Device type: general purpose
Running: Microsoft Windows NT/2K/XP
OS details: Microsoft Windows XP Pro RC1+ through final release
Service Info: OSs: Windows，Windows XP

Nmap finished: 2 IP addresses (2 hosts up) scanned in 88.392 seconds
```

二、 译注

该 Nmap 参考指南中文版由 Fei Yang fyang1024@gmail.com 和 Lei Li lilei_721@6611.org 从 英文版本翻译而来。我们希望这将使全世界使用中文的人们更了解 Nmap，但我们不能保证该译 本和官方的英文版本一样完整，也不能保证同步更新。它可以在 Creative Commons Attribution License 下被修改并重新发布。

三、 选项概要

当 Nmap 不带选项运行时，该选项概要会被输出，最新的版本在这里 http://www.insecure.org/nmap/data/nmap.usage.txt。它帮助人们记住最常用的选项，但不 能替代本手册其余深入的文档，一些晦涩的选项甚至不在这里。

```
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
    Can pass hostnames, IP addresses, networks, etc.
    Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0-255.0-255.1-254
    -iL <inputfilename>: Input from list of hosts/networks
    -iR <num hosts>: Choose random targets
    --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
    --excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
    -sL: List Scan - simply list targets to scan
    -sP: Ping Scan - go no further than determining if host is online
    -P0: Treat all hosts as online -- skip host discovery
    -PS/PA/PU [portlist]: TCP SYN/ACK or UDP discovery probes to given ports
    -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
    -n/-R: Never do DNS resolution/Always resolve [default: sometimes resolve]
SCAN TECHNIQUES:
    -sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans
    -sN/sF/sX: TCP Null, FIN, and Xmas scans
    --scanflags <flags>: Customize TCP scan flags
    -sI <zombie host[:probeport]>: Idlescan
    -sO: IP protocol scan
    -b <ftp relay host>: FTP bounce scan
PORT SPECIFICATION AND SCAN ORDER:
    -p <port ranges>: Only scan specified ports
    Ex: -p22; -p1-65535; -p U:53,111,137,T:21-25,80,139,8080
    -F: Fast - Scan only the ports listed in the nmap-services file)
    -r: Scan ports consecutively - don't randomize
SERVICE/VERSION DETECTION:
    -sV: Probe open ports to determine service/version info
    --version-light: Limit to most likely probes for faster identification
    --version-all: Try every single probe for version detection
    --version-trace: Show detailed version scan activity (for debugging)
OS DETECTION:
    -O: Enable OS detection
    --osscan-limit: Limit OS detection to promising targets
    --osscan-guess: Guess OS more aggressively
TIMING AND PERFORMANCE:
    -T[0-6]: Set timing template (higher is faster)
    --min-hostgroup/max-hostgroup <msec>: Parallel host scan group sizes
    --min-parallelism/max-parallelism <msec>: Probe parallelization
    --min_rtt_timeout/max-rtt-timeout/initial-rtt-timeout <msec>: Specifies probe round t
    --host-timeout <msec>: Give up on target after this long
    --scan-delay/--max_scan-delay <msec>: Adjust delay between probes
FIREWALL/IDS EVASION AND SPOOFING:
    -f; --mtu <val>: fragment packets (optionally w/given MTU)
    -D <decoy1,decoy2[,ME],...>: Cloak a scan with decoys
    -S <IP_Address>: Spoof source address
    -e <iface>: Use specified interface
    -g/--source-port <portnum>: Use given port number
    --data-length <num>: Append random data to sent packets
    --ttl <val>: Set IP time-to-live field
    --spoof-mac <mac address, prefix, or vendor name>: Spoof your MAC address
OUTPUT:
    -oN/-oX/-oS/-oG <file>: Output scan results in normal, XML, s|<rIpt kIddi3, and Grepa
    -oA <basename>: Output in the three major formats at once
    -v: Increase verbosity level (use twice for more effect)
    -d[level]: Set or increase debugging level (Up to 9 is meaningful)
    --packet-trace: Show all packets sent and received
    --iflist: Print host interfaces and routes (for debugging)
```

```
    --append-output: Append to rather than clobber specified output files
    --resume <filename>: Resume an aborted scan
    --stylesheet <path/URL>: XSL stylesheet to transform XML output to HTML
    --no_stylesheet: Prevent Nmap from associating XSL stylesheet w/XML output
MISC:
    -6: Enable IPv6 scanning
    -A: Enables OS detection and Version detection
    --datadir <dirname>: Specify custom Nmap data file location
    --send-eth/--send-ip: Send packets using raw ethernet frames or IP packets
    --privileged: Assume that the user is fully privileged
    -V: Print version number
    -h: Print this help summary page.
EXAMPLES:
    nmap -v -A scanme.nmap.org
    nmap -v -sP 192.168.0.0/16 10.0.0.0/8
    nmap -v -iR 10000 -P0 -p 80
```

四、 目标说明

除了选项，所有出现在 Nmap 命令行上的都被视为对目标主机的说明。最简单的情况是指定一个 目标 IP 地址或主机名。

有时候您希望扫描整个网络的相邻主机。为此，Nmap 支持 CIDR 风格的地址。您可以附加一个

/numbit 在一个 IP 地址或主机名后面， Nmap 将会扫描所有和该参考 IP 地址具有 numbit 相同 比特的所有 IP 地址或主机。例如，192.168.10.0/24 将会扫描 192.168.10.0 (二进制格式: 11000000 10101000 00001010 00000000)和 192.168.10.255 (二进制格式: 11000000 10101000

00001010 11111111)之间的 256 台主机。 192.168.10.40/24 将会做同样的事情。假设主机 scanme.nmap.org 的 IP 地址是 205.217.153.62， scanme.nmap.org/16 将扫描 205.217.0.0 和 205.217.255.255 之间的 65,536 个 IP 地址。所允许的最小值是/1，这将会扫描半个互联网。 最大值是/32，这将会扫描该主机或 IP 地址，因为所有的比特都固定了。

CIDR 标志位很简洁但有时候不够灵活 例如 您也许想要扫描 192.168.0.0/16 但略过任何以.0 或者.255 结束的 IP 地址，因为它们通常是广播地址。 Nmap 通过八位字节地址范围支持这样的 扫描您可以用逗号分开的数字或范围列表为 IP 地址的每个八位字节指定它的范围。例如，192.168.0-255.1-254 将略过在该范围内以.0 和.255 结束的地址。 范围不必限于最后的 8 位：0-255.0-255.13.37 将在整个互联网范围内扫描所有以 13.37 结束的地址。这种大范围的扫描 对互联网调查研究也许有用。

IPv6 地址只能用规范的 IPv6 地址或主机名指定。 CIDR 和八位字节范围不支持 IPv6，因为它 们对于 IPv6 几乎没什么用。

Nmap 命令行接受多个主机说明，它们不必是相同类型。命令 `nmap scanme.nmap.org 192.168.0.0/8 10.0.0，1，3-7.0-255` 将和您预期的一样执行。

虽然目标通常在命令行指定，下列选项也可用来控制目标的选择：

`-iL <inputfilename>` (从列表中输入)

从 inputfilename 中读取目标说明。在命令行输入 一堆主机名显得很笨拙，然而经常需 要这样。 例如，您的 DHCP 服务器可能导出 10,000 个当前租约的列表，而您希望对它们 进行 扫描。如果您不是使用未授权的静态 IP 来定位主机，或许您想要扫描所有 IP 地址 只要生成要扫描的主机的列表，用-iL 把文件名作为选项传给 Nmap。列表中的项可以是 Nmap 在 命令行上接受的任何格式(IP 地址，主机名，CIDR，IPv6，或者八位字节范围) 每一项必须以一个或多个空格，制表符或换行符分开。 如果您希望 Nmap 从标准输入而 不是实际文件读取列表，您可以用一个连字符(-)作为文件名。

`-iR <hostnum>` (随机选择目标)

对于互联网范围内的调查和研究， 您也许想随机地选择目标。 hostnum 选项告诉 Nmap 生成多少个 IP。不合需要的 IP 如特定的私有，组播或者未分配的地址自动 略过。选项 0 意味着永无休止的扫描。记住，一些网管对于未授权的扫描可能会很感冒并加以抱怨。使 用该选项的后果自负! 如果在某个雨天的下午，您觉得实在无聊， 试试这个命令 nmap `-sS -PS80 -iR 0 -p 80` 随机地找一些网站浏览。

`--exclude <host1[，host2][，host3]，...>` (排除主机/网络) 如果在您指定的扫描范围有一些主机或网络不是您的目标， 那就用该选项加上以逗号分隔的列表排除它们。该列表用正常的 Nmap 语法， 因此它可以包括主机名，CIDR，八位字节范围等等。 当您希望扫描的网络包含执行关键任务的服务器，已知的对端口扫描反 应强烈的 系统或者被其它人看管的子网时，这也许有用。

`--excludefile <excludefile>` (排除文件中的列表)

这和--exclude 选项的功能一样，只是所排除的目标是用以 换行符，空格，或者制表符 分隔的 excludefile 提供的，而不是在命令行上输入的。

五、 主机发现

任何网络探测任务的最初几个步骤之一就是把一组 IP 范围(有时该范围是巨大的)缩小为 一列活动的或者您感兴趣的主机。扫描每个 IP 的每个端口很慢，通常也没必要。 当然，什么样的主机令您感兴趣主要依赖于扫描的目的。网管也许只对运行特定服务的 主机感兴趣，而从事安 全的人士则可能对一个马桶都感兴趣，只要它有 IP 地址:-)。一个系统管理员 也许仅仅使用 Ping 来定位内网上的主机 而一个外部入侵测试人员则可能绞尽脑汁用各种方法试图 突破防火 墙的封锁。

由于主机发现的需求五花八门，Nmap 提供了一箩筐的选项来定制您的需求。 主机发现有时候也 叫做 ping 扫描，但它远远超越用世人皆知的 ping 工具 发送简单的 ICMP 回声请求报文。用户 完全可以通过使用列表扫描(-sL)或者 通过关闭 ping (-P0)跳过 ping 的步骤，也可以使用多个端口把 TPC SYN/ACK，UDP 和 ICMP 任意组合起来玩一玩。这些探测的目的是获得响应以显示某 个 IP 地址是否是活动的(正在被某 主机或者网络设备使用)。 在许多网络上，在给定的时间， 往往只有小部分的 IP 地址是活动的。 这种情况在基于 RFC1918 的私有地址空间如 10.0.0.0/8 尤其普遍。 那个网络有 16,000,000 个 IP，但我见过一些使用它的公司连 1000 台机器都没有 主机发现能够找到零星分布于 IP 地址海洋上的那些机器。

如果没有给出主机发现的选项，Nmap 就发送一个 TCP ACK 报文到 80 端口和一个 ICMP 回声请求 到每台目标机器。 一个例外是 ARP 扫描用于局域网上的任何目标机器。对于非特权 UNIX shell 用户，使用 connect()系统调用会发送一个 SYN 报文而不是 ACK 这些默认行为和使用 `-PA -PE` 选项的效果相同。 扫描局域网时，这种主机发现一般够用了，但是对于安全审核，建议进行 更 加全面的探测。

`-P*` 选项(用于选择 ping 的类型)可以被结合使用。 您可以通过使用不同的 TCP 端口/标志位和 ICMP 码发送许多探测报文 来增加穿透防守严密的防火墙的机会 另外要注意的是即使您指定了 其它 `-P*` 选项，ARP 发现(`-PR`)对于局域网上的 目标而言是默认行为，因为它总是更快更有效

下列选项控制主机发现。

`-sL` (列表扫描)

列表扫描是主机发现的退化形式，它仅仅列出指定网络上的每台主机， 不发送任何报文 到目标主机。默认情况下，Nmap 仍然对主机进行反向域名解析以获取 它们的名字。简单 的主机名能给出的有用信息常常令人惊讶。例如， fw.chi.playboy.com 是花花公子芝加 哥办公室的防火墙。Nmap 最后还会报告 IP 地址的总数。列表扫描可以很好的确保您拥 有正确的目标 IP。 如果主机的域名出乎您的意料，那么就值得进一步检查以防错误地扫 描其它组织的网络。

既然只是打印目标主机的列表 像其它一些高级功能如端口扫描 操作系统探测或者 Ping 扫描的选项就没有了。如果您希望关闭 ping 扫描而仍然执行这样的高级功能，请继续 阅读关于 -P0 选项的介绍。

`-sP` (Ping 扫描)

该选项告诉 Nmap 仅仅 进行 ping 扫描 (主机发现)，然后打印出对扫描做出响应的那些 主机。 没有进一步的测试 (如端口扫描或者操作系统探测)。 这比列表扫描更积极，常 常用于和列表扫描相同的目的。它可以得到些许目标网络的信息而不被特别注意到。 对 于攻击者来说 了解多少主机正在运行比列表扫描提供的一列 IP 和主机名往往更有价值

系统管理员往往也很喜欢这个选项。 它可以很方便地得出 网络上有多少机器正在运行 或者监视服务器是否正常运行。常常有人称它为 地毯式 ping，它比 ping 广播地址更可 靠，因为许多主机对广播请求不响应。

`-sP` 选项在默认情况下， 发送一个 ICMP 回声请求和一个 TCP 报文到 80 端口。如果非特权用户执行，就发送一个 SYN 报文 (用 connect()系统调用)到目标机的 80 端口。 当特 权用户扫描局域网上的目标机时，会发送 ARP 请求(`-PR`)， ，除非使用了 `--send-ip` 选项

`-sP` 选项可以和除-P0)之外的任何发现探测类型-P* 选项结合使用以达到更大的灵活性 一旦使用了任何探测类型和端口选项，默认的探测(ACK 和回应请求)就被覆盖了。 当防 守严密的防火墙位于运行 Nmap 的源主机和目标网络之间时， 推荐使用那些高级选项。 否则，当防火墙捕获并丢弃探测包或者响应包时，一些主机就不能被探测到。

`-P0` (无 ping)

该选项完全跳过 Nmap 发现阶段。 通常 Nmap 在进行高强度的扫描时用它确定正在运行的机器。 默认情况下，Nmap 只对正在运行的主机进行高强度的探测如 端口扫描，版本探 测，或者操作系统探测。用-P0 禁止 主机发现会使 Nmap 对每一个指定的目标 IP 地址 进 行所要求的扫描。所以如果在命令行指定一个 B 类目标地址空间(/16)， 所有 65,536 个 IP 地址都会被扫描。 -P0 的第二个字符是数字 0 而不是字母 O。 和列表扫描一样，跳过 正常的主机发现，但不是打印一个目标列表， 而是继续执行所要求的功能，就好像每个 IP 都是活动的。

`-PS [portlist]` (TCP SYN Ping)

该选项发送一个设置了 SYN 标志位的空 TCP 报文。 默认目的端口为 80 (可以通过改变 nmap.h) 文件中的 DEFAULT_TCP_PROBE_PORT 值进行配置，但不同的端口也可以作为选项 指定。 甚至可以指定一个以逗号分隔的端口列表(如 `-PS22，23，25，80，113，1050， 35000`)， 在这种情况下，每个端口会被并发地扫描。

SYN 标志位告诉对方您正试图建立一个连接。 通常目标端口是关闭的，一个 RST (复位) 包会发回来。 如果碰巧端口是开放的，目标会进行 TCP 三步握手的第二步，回应 一个 SYN/ACK TCP 报文。然后运行 Nmap 的机器则会扼杀这个正在建立的连接， 发送一个 RST 而非 ACK 报文，否则，一个完全的连接将会建立。 RST 报文是运行 Nmap 的机器而不是 Nmap 本身响应的，因为它对收到 的 SYN/ACK 感到很意外。

Nmap 并不关心端口开放还是关闭。 无论 RST 还是 SYN/ACK 响应都告诉 Nmap 该主机正在运行。

在 UNIX 机器上，通常只有特权用户 root 能否发送和接收 原始的 TCP 报文。因此作为 一个变通的方法，对于非特权用户， Nmap 会为每个目标主机进行系统调用 connect()， 它也会发送一个 SYN 报文来尝试建立连接。如果 connect()迅速返回成功或者一个 ECONNREFUSED 失败，下面的 TCP 堆栈一定已经收到了一个 SYN/ACK 或者 RST，该主机将 被 标志位为在运行。 如果连接超时了，该主机就标志位为 down 掉了。这种方法也用于 IPv6 连接，因为 Nmap 目前还不支持原始的 IPv6 报文。

`-PA [portlist]` (TCP ACK Ping)

TCP ACK ping 和刚才讨论的 SYN ping 相当类似。 也许您已经猜到了，区别就是设置 TCP 的 ACK 标志位而不是 SYN 标志位。 ACK 报文表示确认一个建立连接的尝试，但该连接尚未完全建立。 所以远程主机应该总是回应一个 RST 报文， 因为它们并没有发出过连接 请求到运行 Nmap 的机器，如果它们正在运行的话。

`-PA` 选项使用和 SYN 探测相同的默认端口(80) 也可以 用相同的格式指定目标端口列表 如果非特权用户尝试该功能， 或者指定的是 IPv6 目标，前面说过的 connect()方法将被 使用。 这个方法并不完美，因为它实际上发送的是 SYN 报文，而不是 ACK 报文。

提供 SYN 和 ACK 两种 ping 探测的原因是使通过防火墙的机会尽可能大。 许多管理员会 配置他们的路由器或者其它简单的防火墙来封锁 SYN 报文，除非 连接目标是那些公开的 服务器像公司网站或者邮件服务器。 这可以阻止其它进入组织的连接，同时也允许用户 访问互联网。 这种无状态的方法几乎不占用防火墙/路由器的资源，因而被硬件和软件 过滤器 广泛支持。Linux Netfilter/iptables 防火墙软件提供方便的 --syn 选项来实 现这种无状态的方法。 当这样的无状态防火墙规则存在时，发送到关闭目标端口的 SYN ping 探测 (-PS) 很可能被封锁。这种情况下，ACK 探测格外有闪光点，因为它正好利用 了 这样的规则。

另外一种常用的防火墙用有状态的规则来封锁非预期的报文。 这一特性已开始只存在于 高端防火墙，但是这些年类它越来越普遍了。 Linux Netfilter/iptables 通过 `--state` 选项支持这一特性，它根据连接状态把报文 进行分类。SYN 探测更有可能用于这样的系 统，由于没头没脑的 ACK 报文 通常会被识别成伪造的而丢弃。解决这个两难的方法是通 过即指定 `-PS` 又指定`-PA` 来即发送 SYN 又发送 ACK。

`-PU [portlist]` (UDP Ping)

还有一个主机发现的选项是 UDP ping，它发送一个空的(除非指定了--data-length UDP 报文到给定的端口。端口列表的格式和前面讨论过的-PS 和-PA 选项还是一样。 如果不 指定端口，默认是 31338。该默认值可以通过在编译时改变 nmap.h 文件中的 DEFAULT_UDP_PROBE_PORT 值进行配置。默认使用这样一个奇怪的端口是因为对开放端口 进行这种扫描一般都不受欢迎。

如果目标机器的端口是关闭的，UDP 探测应该马上得到一个 ICMP 端口无法到达的回应报文。 这对于 Nmap 意味着该机器正在运行。 许多其它类型的 ICMP 错误，像主机/网络无 法到达或者 TTL 超时则表示 down 掉的或者不可到达的主机。 没有回应也被这样解释。 如果到达一个开放的端口，大部分服务仅仅忽略这个 空报文而不做任何回应。这就是为 什么默认探测端口是 31338 这样一个 极不可能被使用的端口。少数服务如 chargen 会响 应一个空的 UDP 报文， 从而向 Nmap 表明该机器正在运行。

该扫描类型的主要优势是它可以穿越只过滤 TCP 的防火墙和过滤器。 例如。我曾经有过 一个 Linksys BEFW11S4 无线宽带路由器。默认情况下， 该设备对外的网卡过滤所有 TCP 端口，但 UDP 探测仍然会引发一个端口不可到达 的消息，从而暴露了它自己。

`-PE; -PP; -PM` (ICMP Ping Types)

除了前面讨论的这些不常见的 TCP 和 UDP 主机发现类型 Nmap 也能发送世人皆知的 ping 程序所发送的报文。Nmap 发送一个 ICMP type 8 (回声请求)报文到目标 IP 地址， 期待 从运行的主机得到一个 type 0 (回声响应)报文。 对于网络探索者而言，不幸的是，许 多主机和 防火墙现在封锁这些报文，而不是按期望的那样响应， 参见 RFC 1122。因此

仅仅 ICMP 扫描对于互联网上的目标通常是不够的。 但对于系统管理员监视一个内部网 络，它们可能是实际有效的途径。 使用-PE 选项打开该回声请求功能。

虽然回声请求是标准的 ICMP ping 查询， Nmap 并不止于此。ICMP 标准 (RFC 792)还规 范了时间戳请求，信息请求 request，和地址掩码请求，它们的代码分别是 13，15 和 17 虽然这些查询的表面目的是获取信息如地址掩码和当前时间， 它们也可以很容易地用于 主机发现。 很简单，回应的系统就是在运行的系统。Nmap 目前没有实现信息请求报文 因为它们还没有被广泛支持。RFC 1122 坚持“主机不应该实现这些消息”。 时间戳和 地址掩码查询可以分别用-PP 和-PM 选项发送。 时间戳响应(ICMP 代码 14)或者地址掩码 响应(代码 18)表示主机在运行。 当管理员特别封锁了回声请求报文而忘了其它 ICMP 查 询可能用于 相同目的时，这两个查询可能很有价值。

`-PR` (ARP Ping)

最常见的 Nmap 使用场景之一是扫描一个以太局域网。 在大部分局域网上，特别是那些 使用基于 RFC1918 私有地址范围的网络，在一个给定的时间绝大部分 IP 地址都是不使 用的。 当 Nmap 试图发送一个原始 IP 报文如 ICMP 回声请求时， 操作系统必须确定对应 于目标 IP 的硬件 地址(ARP)，这样它才能把以太帧送往正确的地址。 这一般比较慢而 且会有些问题，因为操作系统设计者认为一般不会在短时间内 对没有运行的机器作几百 万次的 ARP 请求。

当进行 ARP 扫描时，Nmap 用它优化的算法管理 ARP 请求。 当它收到响应时， Nmap 甚至不需要担心基于 IP 的 ping 报文，既然它已经知道该主机正在运行了。 这使得 ARP 扫描 比基于 IP 的扫描更快更可靠。 所以默认情况下，如果 Nmap 发现目标主机就在它所在的 局域网上，它会进行 ARP 扫描。 即使指定了不同的 ping 类型(如 -PI 或者 -PS) ，Nmap 也会对任何相同局域网上的目标机使用 ARP。 如果您真的不想要 ARP 扫描，指定 `--send-ip` 。

`-n` (不用域名解析)

告诉 Nmap 永不对它发现的活动 IP 地址进行反向域名解析。 既然 DNS 一般比较慢，这可 以让事情更快些。

`-R` (为所有目标解析域名)

告诉 Nmap 永远 对目标 IP 地址作反向域名解析。 一般只有当发现机器正在运行时才进 行这项操作。

`--system_dns` (使用系统域名解析器)

默认情况下，Nmap 通过直接发送查询到您的主机上配置的域名服务器 来解析域名。为了 提高性能，许多请求 (一般几十个) 并发执行。如果您希望使用系统自带的解析器，就 指定该选项 (通过 getnameinfo()调用一次解析一个 IP) 除非 Nmap 的 DNS 代码有 bug-- 如果是这样，请联系我们。 一般不使用该选项，因为它慢多了。系统解析器总是用于 IPv6 扫描。

六、 端口扫描基础

虽然 Nmap 这些年来功能越来越多， 它也是从一个高效的端口扫描器开始的，并且那仍然是它 的核心功能。 nmap target 这个简单的命令扫描主机 target 上的超过 1660 个 TCP 端口。。 许多传统的端口扫描器只列出所有端口是开放还是关闭的， Nmap 的信息粒度比它们要细得多 它把端口分成六个状态: open(开放的)， closed(关闭的)，filtered(被过滤的)，unfiltered(未被过滤的)， open|filtered(开放或者被过滤的)，或者 closed|filtered(关闭 或者未被过滤的)。

这些状态并非端口本身的性质，而是描述 Nmap 怎样看待它们。例如，对于同样的目标机器的 135/tcp 端口，从同网络扫描显示它是开放的，而跨网络作完全相同的扫描则可能显示它是 filtered(被过滤的)。

Nmap 所识别的 6 个端口状态。 open(开放的)

应用程序正在该端口接收 TCP 连接或者 UDP 报文 发现这一点常常是端口扫描 的主要目标。安全意识强的人们知道每个开放的端口 都是攻击的入口。攻击者或者入侵测试者想 要发现开放的端口。 而管理员则试图关闭它们或者用防火墙保护它们以免妨碍了合法用 户。 非安全扫描可能对开放的端口也感兴趣，因为它们显示了网络上那些服务可供使用

closed(关闭的)

关闭的端口对于 Nmap 也是可访问的(它接受 Nmap 的探测报文并作出响应)， 但没有应用 程序在其上监听。 它们可以显示该 IP 地址上(主机发现，或者 ping 扫描)的主机正在运 行 up 也对部分操作系统探测有所帮助。 因为关闭的关口是可访问的，也许过会儿值得 再扫描一下，可能一些又开放了。 系统管理员可能会考虑用防火墙封锁这样的端口。 那 样他们就会被显示为被过滤的状态，下面讨论。

filtered(被过滤的)

由于包过滤阻止探测报文到达端口， Nmap 无法确定该端口是否开放。过滤可能来自专业 的防火墙设备，路由器规则 或者主机上的软件防火墙。这样的端口让攻击者感觉很挫折 因为它们几乎不提供 任何信息。有时候它们响应 ICMP 错误消息如类型 3 代码 13 (无法 到达目标: 通信被管理员禁止)，但更普遍的是过滤器只是丢弃探测帧， 不做任何响应 这迫使 Nmap 重试若干次以访万一探测包是由于网络阻塞丢弃的。 这使得扫描速度明显 变慢。

unfiltered(未被过滤的)

未被过滤状态意味着端口可访问，但 Nmap 不能确定它是开放还是关闭。 只有用于映射 防火墙规则集的 ACK 扫描才会把端口分类到这种状态。 用其它类型的扫描如窗口扫描， SYN 扫描，或者 FIN 扫描来扫描未被过滤的端口可以帮助确定 端口是否开放。

open|filtered(开放或者被过滤的)

当无法确定端口是开放还是被过滤的，Namp 就把该端口划分成 这种状态。开放的端口不 响应就是一个例子。没有响应也可能意味着报文过滤器丢弃 了探测报文或者它引发的任 何响应。因此 Nmap 无法确定该端口是开放的还是被过滤的。 UDP，IP 协议， FIN，Null 和 Xmas 扫描可能把端口归入此类。

closed|filtered(关闭或者被过滤的)

该状态用于 Nmap 不能确定端口是关闭的还是被过滤的。 它只可能出现在 IPID Idle 扫 描中。

七、 端口扫描技术

作为一个修车新手，我可能折腾几个小时来摸索怎样把基本工具(锤子，胶带，扳子等) 用于手 头的任务。当我惨痛地失败，把我的老爷车拖到一个真正的技师那儿的时候，他总是在他的工 具箱里翻来翻去，直到拽出一个完美的工具然后似乎不费吹灰之力搞定它。端口扫描的艺术和 这个类似。专家理解成打的扫描技术，选择最适合的一种 (或者组合)来完成给定的任务。 另 一方面，没有经验的用户和刚入门者总是用默认的 SYN 扫描解决每个问题。既然 Nmap 是免费的 掌握端口扫描的唯一障碍就是知识。这当然是汽车世界所不能比的，在那里，可能需要高超的 技巧才能确定您需要一个压杆弹簧压缩机，接着您还得为它付数千美金。

大部分扫描类型只对特权用户可用。这是因为他们发送接收原始报文，这在 Unix 系统需要 root 权限。在 Windows 上推荐使用 administrator 账户，但是当 WinPcap 已经被加载到操作系统时 非特权用户也可以正常使用 Nmap 当 Namp 在 1997 年发布时 需要 root 权限是一个严重的局限 因为很多用户只有共享的 shell 账户。现在，世界变了，计算机便宜了，更多人拥有互联网连 接，桌面 UNIX 系统 (包括 Linux 和 MAC OS X)很普遍了。Windows 版本的 Nmap 现在也有了，这 使它可以运行在更多的桌面上。由于所有这些原因，用户不再需要用有限的共享 shell 账户运 行 Nmap。这是很幸运的，因为特权选项让 Nmap 强大得多也灵活得多。

虽然 Nmap 努力产生正确的结果，但请记住所有结果都是基于目标机器(或者它们前面的防火墙) 返回的报文的。。这些主机也许是不值得信任的，它们可能响应以迷惑或误导 Nmap 的报文。更 普遍的是非 RFC 兼容的主机以不正确的方式响应 Nmap 探测。FIN，Null 和 Xmas 扫描特别容易遇 到这个问题。这些是特定扫描类型的问题，因此我们在个别扫描类型里讨论它们。

这一节讨论 Nmap 支持的大约十几种扫描技术。一般一次只用一种方法，除了 UDP 扫描(-sU)可 能和任何一种 TCP 扫描类型结合使用 友情提示一下 端口扫描类型的选项格式是-sC 其中 C 是 个显眼的字符，通常是第一个字符。一个例外是 deprecated FTP bounce 扫描(-b)。默认情况 下，Nmap 执行一个 SYN 扫描，但是如果用户没有权限发送原始报文(在 UNIX 上需要 root 权限) 或者如果指定的是 IPv6 目标，Nmap 调用 connect()。本节列出的扫描中，非特权用户只能执行 connect()和 ftp bounce 扫描。

`-sS` (TCP SYN 扫描)

SYN 扫描作为默认的也是最受欢迎的扫描选项，是有充分理由的。它执行得很快，在一个 没有入侵防火墙的快速网络上，每秒钟可以扫描数千个端口。 SYN 扫描相对来说不张扬 不易被注意到，因为它从来不完成 TCP 连接。它也不像 Fin/Null/Xmas，Maimon 和 Idle 扫描依赖于特定平台，而可以应对任何兼容的 TCP 协议栈。它还可以明确可靠地区分 open(开放的)，closed(关闭的)，和 filtered(被过滤的) 状态

它常常被称为半开放扫描，因为它不打开一个完全的 TCP 连接。它发送一个 SYN 报文， 就像您真的要打开一个连接，然后等待响应。 SYN/ACK 表示端口在监听 (开放)，而 RST (复位)表示没有监听者。如果数次重发后仍没响应，该端口就被标记为被过滤。如果收 到 ICMP 不可到达错误 (类型 3，代码 1，2，3，9，10，或者 13)，该端口也被标记为被 过滤。

`-sT` (TCP connect()扫描)

当 SYN 扫描不能用时，CP Connect()扫描就是默认的 TCP 扫描。当用户没有权限发送原 始报文或者扫描 IPv6 网络时，就是这种情况。 Instead of writing raw packets as most other scan types do，Nmap 通过创建 connect() 系统调用要求操作系统和目标机以及 端口建立连接，而不像其它扫描类型直接发送原始报文。 这是和 Web 浏览器，P2P 客户 端以及大多数其它网络应用程序用以建立连接一样的 高层系统调用 它是叫做 Berkeley Sockets API 编程接口的一部分。Nmap 用该 API 获得每个连接尝试的状态信息，而不是 读取响应的原始报文。

当 SYN 扫描可用时，它通常是更好的选择。因为 Nmap 对高层的 connect()调用比对原始 报文控制更少 所以前者效率较低 该系统调用完全连接到开放的目标端口而不是像 SYN 扫描进行半开放的复位。这不仅花更长时间，需要更多报文得到同样信息，目标机也更 可能记录下连接。IDS(入侵检测系统)可以捕获两者，但大部分机器没有这样的警报系统 当 Nmap 连接，然后不发送数据又关闭连接，许多普通 UNIX 系统上的服务会在 syslog 留 下记录，有时候是一条加密的错误消息。此时，有些真正可怜的服务会崩溃，虽然这不 常发生。如果管理员在日志里看到来自同一系统的一堆连接尝试，她应该知道她的系统 被扫描了。

`-sU` (UDP 扫描)

虽然互联网上很多流行的服务运行在 TCP 协议上，UDP 服务也不少。 DNS，SNMP，和 DHCP (注册的端口是 53，161/162，和 67/68)是最常见的三个。因为 UDP 扫描一般较慢，比 TCP 更困难，一些安全审核人员忽略这些端口。这是一个错误，因为可探测的 UDP 服务相当 普遍，攻击者当然不会忽略整个协议。所幸，Nmap 可以帮助记录并报告 UDP 端口。

UDP 扫描用 `-sU` 选项激活 它可以和 TCP 扫描如 SYN 扫描 (`-sS`)结合使用来同时检查两种协议。

UDP 扫描发送空的(没有数据)UDP 报头到每个目标端口。如果返回 ICMP 端口不可到达错 误(类型 3，代码 3)，该端口是 closed(关闭的)。 其它 ICMP 不可到达错误(类型 3，代 码 1，2，9，10，或者 13)表明该端口是 filtered(被过滤的)。偶尔地，某服务会响应一 个 UDP 报文，证明该端口是 open(开放的)。如果几次重试后还没有响应，该端口就被认为是 open|filtered(开放|被过滤的)。这意味着该端口可能是开放的，也可能包过滤器 正在封锁通信。可以用版本扫描(-sV)帮助区分真正的开放端口和被过滤的端口。

UDP 扫描的巨大挑战是怎样使它更快速。开放的和被过滤的端口很少响应，让 Nmap 超时 然后再探测，以防探测帧或者响应丢失。关闭的端口常常是更大的问题。它们一般发回 一个 ICMP 端口无法到达错误 但是不像关闭的 TCP 端口响应 SYN 或者 Connect 扫描所发 送的

RST 报文，许多主机在默认情况下限制 ICMP 端口不可到达消息。 Linux 和 Solaris 对此特别严格。例如， Linux 2.4.20 内核限制一秒钟只发送一条目标不可到达消息 (见 net/ipv4/icmp。c)。

Nmap 探测速率限制并相应地减慢来避免用那些目标机会丢弃的无用报文来阻塞网络。不 幸的是，Linux 式的一秒钟一个报文的限制使 65,536 个端口的扫描要花 18 小时以上。 加速 UDP 扫描的方法包括并发扫描更多的主机，先只对主要端口进行快速扫描，从防火 墙后面扫描，使用--host-timeout 跳过慢速的主机。

`-sN; -sF; -sX` (TCP Null，FIN，and Xmas 扫描)

这三种扫描类型 (甚至用下一节描述的 --scanflags 选项的更多类型) 在 TCP RFC 中 发掘了一个微妙的方法来区分 open(开放的)和 closed(关闭的)端口。第 65 页说“如果 [目标]端口状态是关闭的.... 进入的不含 RST 的报文导致一个 RST 响应。” 接下来的 一页讨论不设置 SYN，RST，或者 ACK 位的报文发送到开放端口: “理论上，这不应该发 生，如果您确实收到了，丢弃该报文，返回。 ”

如果扫描系统遵循该 RFC，当端口关闭时，任何不包含 SYN，RST，或者 ACK 位的报文会导致一个 RST 返回，而当端口开放时，应该没有任何响应。只要不包含 SYN，RST，或者 ACK，任何其它三种(FIN，PSH，and URG)的组合都行。Nmap 有三种扫描类型利用这一点

Null 扫描 (`-sN`) 不设置任何标志位(tcp 标志头是 0) FIN 扫描 (`-sF`)

只设置 TCP FIN 标志位。 Xmas 扫描 (-sX)

设置 FIN，PSH，和 URG 标志位，就像点亮圣诞树上所有的灯一样。

除了探测报文的标志位不同，这三种扫描在行为上完全一致。如果收到一个 RST 报文， 该端口被认为是 closed(关闭的) 而没有响应则意味着端口是 open|filtered(开放或者 被过滤的)。如果收到 ICMP 不可到达错误(类型 3，代号 1，2，3，9，10，或者 13)，该 端口就被标记为被过滤的。

这些扫描的关键优势是它们能躲过一些无状态防火墙和报文过滤路由器。另一个优势是 这些扫描类型甚至比 SYN 扫描还要隐秘一些。但是别依赖它 -- 多数现代的 IDS 产品可

以发现它们。一个很大的不足是并非所有系统都严格遵循 RFC 793。 许多系统不管端口 开放还是关闭，都响应 RST。这导致所有端口都标记为 closed(关闭的)。这样的操作系 统主要有 Microsoft Windows，许多 Cisco 设备，BSDI，以及 IBM OS/400。但是这种扫 描对多数 UNIX 系统都能工作 这些扫描的另一个不足是它们不能辨别 open(开放的)端口 和一些特定的 filtered(被过滤的)端口，从而返回 open|filtered(开放或者被过滤 的)。

`-sA` (TCP ACK 扫描)

这种扫描与目前为止讨论的其它扫描的不同之处在于它不能确定 open(开放的)或者 open|filtered(开放或者过滤的))端口。它用于发现防火墙规则，确定它们是有状态的 还是无状态的，哪些端口是被过滤的。

ACK 扫描探测报文只设置 ACK 标志位(除非您使用 --scanflags) 当扫描未被过滤的系统 时，open(开放的)和 closed(关闭的) 端口都会返回 RST 报文。Nmap 把它们标记为 unfiltered(未被过滤的)，意思是 ACK 报文不能到达，但至于它们是 open(开放的)或者 closed(关闭的) 无法确定。不响应的端口或者发送特定的 ICMP 错误消息(类型 3，代号 1，2，3，9，10，或者 13)的端口，标记为 filtered(被过滤的)。

`-sW` (TCP 窗口扫描)

除了利用特定系统的实现细节来区分开放端口和关闭端口，当收到 RST 时不总是打印 unfiltered，窗口扫描和 ACK 扫描完全一样。它通过检查返回的 RST 报文的 TCP 窗口域 做到这一点。在某些系统上，开放端口用正数表示窗口大小(甚至对于 RST 报文) 而关闭 端口的窗口大小为 0。因此，当收到 RST 时，窗口扫描不总是把端口标记为 unfiltered 而是根据 TCP 窗口值是正数还是 0，分别把端口标记为 open 或者 closed

该扫描依赖于互联网上少数系统的实现细节，因此您不能永远相信它。不支持它的系统 会通常返回所有端口 closed。当然，一台机器没有开放端口也是有可能的。如果大部分 被扫描的端口是 closed，而一些常见的端口 (如 22， 25，53) 是 filtered，该系统 就非常可疑了。偶尔地，系统甚至会显示恰恰相反的行为。 如果您的扫描显示 1000 个 开放的端口和 3 个关闭的或者被过滤的端口，那么那 3 个很可能也是开放的端口。

`-sM` (TCP Maimon 扫描)

Maimon 扫描是用它的发现者 Uriel Maimon 命名的。他在 Phrack Magazine issue #49 (November 1996)中描述了这一技术。 Nmap 在两期后加入了这一技术。这项技术和 Null FIN，以及 Xmas 扫描完全一样，除了探测报文是 FIN/ACK。根据 RFC 793 (TCP)，无论端口开放或者关闭，都应该对这样的探测响应 RST 报文。然而，Uriel 注意到如果端口开放 许多基于 BSD 的系统只是丢弃该探测报文。

`--scanflags` (定制的 TCP 扫描)

真正的 Nmap 高级用户不需要被这些现成的扫描类型束缚 --scanflags 选项允许您通过 指定任意 TCP 标志位来设计您自己的扫描。让您的创造力流动，躲开那些仅靠本手册添 加规则的入侵检测系统！

`--scanflags` 选项可以是一个数字标记值如 9 (PSH 和 FIN)，但使用字符名更容易些。只 要是 URG， ACK，PSH， RST，SYN，and FIN 的任何组合就行。例如，--scanflags URGACKPSHRSTSYNFIN 设置了所有标志位，但是这对扫描没有太大用处。标志位的顺序不 重要。

除了设置需要的标志位，您也可以设置 TCP 扫描类型(如-sA 或者-sF)。那个基本类型告 诉 Nmap 怎样解释响应。例如， SYN 扫描认为没有响应意味着 filtered 端口，而 FIN 扫 描则认为是 open|filtered。除了使用您指定的 TCP 标记位，Nmap 会和基本扫描类型一 样工作。如果您不指定基本类型，就使用 SYN 扫描。

`-sI <zombie host[:probeport]>` (Idlescan)

这种高级的扫描方法允许对目标进行真正的 TCP 端口盲扫描 (意味着没有报文从您的真 实 IP 地址发送到目标)。相反，side-channel 攻击 利用 zombie 主机上已知的 IP 分段 ID 序列生成算法来窥探目标上开放端口的信息。 IDS 系统将显示扫描来自您指定的 zombie 机(必须运行并且符合一定的标准)。 这种奇妙的扫描类型太复杂了，不能在此完 全描述，所以我写一篇非正式的论文， 发布在 http://nmap.org/book/idlescan.html。

除了极端隐蔽(由于它不从真实 IP 地址发送任何报文)， 该扫描类型可以建立机器间的 基于 IP 的信任关系。 端口列表从 zombie 主机的角度。显示开放的端口。 因此您可以 尝试用您认为(通过路由器/包过滤规则)可能被信任的 zombies 扫描目标。

如果您由于 IPID 改变希望探测 zombie 上的特定端口， 您可以在 zombie 主机后加上一 个冒号和端口号。 否则 Nmap 会使用默认端口(80)。

`-sO` (IP 协议扫描)

IP 协议扫描可以让您确定目标机支持哪些 IP 协议 (TCP，ICMP，IGMP，等等)。从技术 上说，这不是端口扫描，既然它遍历的是 IP 协议号而不是 TCP 或者 UDP 端口号。但是它 仍使用 -p 选项选择要扫描的协议号，用正常的端口表格式报告结果，甚至用和真正的端 口扫描一样的扫描引擎。因此它和端口扫描非常接近，也被放在这里讨论。

除了本身很有用，协议扫描还显示了开源软件的力量。尽管基本想法非常简单，我过去 从没想过增加这一功能也没收到任何对它的请求。在 2000 年夏天，Gerhard Rieger 孕育 了这个想法，写了一个很棒的补丁程序，发送到 nmap-hackers 邮件列表。我把那个补丁 加入了 Nmap，第二天发布了新版本。几乎没有商业软件会有用户有足够的热情设计并贡 献他们的改进。

协议扫描以和 UDP 扫描类似的方式工作。它不是在 UDP 报文的端口域上循环，而是在 IP 协议域的 8 位上循环，发送 IP 报文头。报文头通常是空的，不包含数据，甚至不包含所 申明的协议的正确报文头 TCP，UDP，和 ICMP 是三个例外。它们三个会使用正常的协议头，因为否则某些系统拒绝发送，而且 Nmap 有函数创建它们。协议扫描不是注意 ICMP 端口不可到达消息，而是 ICMP 协议不可到达消息。如果 Nmap 从目标主机收到任何协议 的任何响应，Nmap 就把那个协议标记为 open。 ICMP 协议不可到达错误(类型 3，代号 2) 导致协议被标记为 closed。其它 ICMP 不可到达协议(类型 3，代号 1，3，9，10，或者 13) 导致协议被标记为 filtered (虽然同时他们证明 ICMP 是 open)。如果重试之后仍 没有收到响应，该协议就被标记为 open|filtered

`-b <ftp relay host>` (FTP 弹跳扫描)

FTP 协议的一个有趣特征(RFC 959) 是支持所谓代理 ftp 连接。它允许用户连接到一台 FTP 服务器，然后要求文件送到一台第三方服务器。这个特性在很多层次上被滥用，所以 许多服务器已经停止支持它了。其中一种就是导致 FTP 服务器对其它主机端口扫描。只 要请求 FTP 服务器轮流发送一个文件到目标主机上的所感兴趣的端口。错误消息会描述 端口是开放还是关闭的。这是绕过防火墙的好方法，因为 FTP 服务器常常被置于可以访 问比 Web 主机更多其它内部主机的位置。 Nmap 用-b 选项支持 ftp 弹跳扫描。参数格式 是

username:password@server:port。 Server 是某个脆弱的 FTP 服务器的名字或者 IP 地址。您也许可以省略 username:password，如果服务器上开放了匿名用户 (user:anonymous password:-wwwuser@)。端口号(以及前面的冒号) 也可以省略，如果 server 使用默认的 FTP 端口(21)。

当 Nmap1997 年发布时，这个弱点被广泛利用，但现在大部分已经被 fix 了。脆弱的服务 器仍然存在，所以如果其它都失败了，这也值得一试。如果您的目标是绕过防火墙，扫 描目标网络上的开放的 21 端口(或者甚至任何 ftp 服务，如果您用版本探测扫描所有端 口)，然后对每个尝试弹跳扫描。Nmap 会告诉您该主机脆弱与否。如果您只是试着玩 Nmap， 您不必(事实上，不应该)限制您自己。在您随机地在互联网上寻找脆弱的 FTP 服务器时 考虑一下系统管理员不太喜欢您这样滥用他们的服务器。

八、 端口说明和扫描顺序

除了所有前面讨论的扫描方法 Nmap 提供选项说明那些端口被扫描以及扫描是随机还是顺序进 行。 默认情况下，Nmap 用指定的协议对端口 1 到 1024 以及 nmap-services 文件中列出的更高 的端口在扫描。

`-p <port ranges>` (只扫描指定的端口)

该选项指明您想扫描的端口，覆盖默认值。 单个端口和用连字符表示的端口范围(如 1-1023)都可以。 范围的开始以及/或者结束值可以被省略， 分别导致 Nmap 使用 1 和 65535。所以您可以指定 -p-从端口 1 扫描到 65535。 如果您特别指定，也可以扫描端口 0。 对于 IP 协议扫描(-sO)，该选项指定您希望扫描的协议号 (0-255)。

当既扫描 TCP 端口又扫描 UDP 端口时，您可以通过在端口号前加上 T: 或者 U:指定协议 协议限定符一直有效您直到指定另一个。 例如，参数 -p U:53，111，137，T:21-25， 80，139，8080 将扫描 UDP 端口 53，111，和 137，同时扫描列出的 TCP 端口。注意，要既扫描 UDP 又扫描 TCP，您必须指定 -sU ，以及至少一个 TCP 扫描类型(如 -sS，-sF， 或者 -sT)。如果没有给定协议限定符， 端口号会被加到所有协议列表。

`-F` (快速 (有限的端口) 扫描)

在 nmap 的 nmap-services 文件中(对于-sO，是协议文件)指定您想要扫描的端口。 这比 扫描所有 65535 个端口快得多。 因为该列表包含如此多的 TCP 端口(1200 多)，这和默认 的 TCP 扫描 scan (大约 1600 个端口)速度差别不是很大。如果您用--datadir 选项指定 您自己的小小的 nmap-services 文件 ，差别会很惊人。

`-r` (不要按随机顺序扫描端口)

默认情况下，Nmap 按随机顺序扫描端口 (除了出于效率的考虑，常用的端口前移)。这种 随机化通常都是受欢迎的， 但您也可以指定-r 来顺序端口扫描。

九、 服务和版本探测

把 Nmap 指向一个远程机器，它可能告诉您 端口 25/tcp，80/tcp，和 53/udp 是开放的。使用包 含大约 2,200 个著名的服务的 nmap-services 数据库， Nmap 可以报告那些端口可能分别对应 于一个邮件服务器 (SMTP) web 服务器(HTTP) 和域名服务器(DNS) 这种查询通常是正确的 -- 事实上，绝大多数在 TCP 端口 25 监听的守护进程是邮件 服务器。然而，您不应该把赌注押在 这上面! 人们完全可以在一些奇怪的端口上运行服务。

即使 Nmap 是对的，假设运行服务的确实是 SMTP，HTTP 和 DNS，那也不是特别多的信息。 当为 您的公司或者客户作安全评估(或者甚至简单的网络明细清单)时， 您确实想知道正在运行什么 邮件和域名服务器以及它们的版本。 有一个精确的版本号对了解服务器有什么漏洞有巨大帮 助。 版本探测可以帮您获得该信息。

在用某种其它类型的扫描方法发现 TCP 和/或者 UDP 端口后， 版本探测会询问这些端口，确定 到底什么服务正在运行。 nmap-service-probes 数据库包含查询不同服务的探测报文 和解析 识别响应的匹配表达式。 Nmap 试图确定服务协议 (如 ftp，ssh，telnet，http)，应用程序名 (如 ISC Bind，Apache httpd，Solaris telnetd)，版本号， 主机名，设备类型(如 打印机，路由器)，操作系统家族 (如 Windows，Linux)以及其它的细节，如 如是否可以连接 X server SSH 协议版本 ，或者 KaZaA 用户名)。当然，并非所有服务都提供所有这些信息。 如果 Nmap 被编译成支持 OpenSSL， 它将连接到 SSL 服务器，推测什么服务在加密层后面监听。 当发现 RPC 服务时， Nmap RPC grinder (-sR)会自动被用于确定 RPC 程序和它的版本号。 如果在扫描 某个 UDP 端口后仍然无法确定该端口是开放的还是被过滤的，那么该端口状态就 被标记为 open|filtered。 版本探测将试图从这些端口引发一个响应(就像它对开放端口做的一样)， 如 果成功，就把状态改为开放。 open|filtered TCP 端口用同样的方法对待。 注意 Nmap -A 选项 在其它情况下打开版本探测。 有一篇关于版本探测的原理，使用和定制的文章在 http://www.insecure.org/nmap/vscan/。

当 Nmap 从某个服务收到响应，但不能在数据库中找到匹配时， 它就打印一个特殊的 fingerprint 和一个 URL 给您提交，如果您确实知道什么服务运行在端口。 请花两分钟提交您的发现，让每个人受益。由于这些提交， Nmap 有 350 种以上协议如 smtp，ftp，http 等的大约 3，000 条模式匹配。

用下列的选项打开和控制版本探测。

`-sV` (版本探测)

打开版本探测。 您也可以用-A 同时打开操作系统探测和版本探测。

`--allports` (不为版本探测排除任何端口)

默认情况下，Nmap 版本探测会跳过 9100 TCP 端口，因为一些打印机简单地打印送到该端口的 任何数据，这回导致数十页 HTTP get 请求，二进制 SSL 会话请求等等被打印出来 这一行为可以通过修改或删除 nmap-service-probes 中的 Exclude 指示符改变 您也可 以不理会任

何 Exclude 指示符，指定--allports 扫描所有端口

`--version-intensity <intensity>` (设置 版本扫描强度)

当进行版本扫描(-sV)时，nmap 发送一系列探测报文 ，每个报文都被赋予一个 1 到 9 之 间的值。 被赋予较低值的探测报文对大范围的常见服务有效，而被赋予较高值的报文 一 般没什么用。强度水平说明了应该使用哪些探测报文。数值越高， 服务越有可能被正确 识别。 然而，高强度扫描花更多时间。强度值必须在 0 和 9 之间。 默认是 7。当探测报 文通过 nmap-service-probes ports 指示符 注册到目标端口时，无论什么强度水平，探 测报文都会被尝试。这保证了 DNS 探测将永远在任何开放的 53 端口尝试， SSL 探测将 在 443 端口尝试，等等。

`--version-light` (打开轻量级模式)

这是 --version-intensity 2 的方便的别名。轻量级模式使 版本扫描快许多，但它识别 服务的可能性也略微小一点。

`--version-all` (尝试每个探测)

`--version-intensity` 9 的别名， 保证对每个端口尝试每个探测报文。

`--version-trace` (跟踪版本扫描活动)

这导致 Nmap 打印出详细的关于正在进行的扫描的调试信息。 它是您用--packet-trace 所得到的信息的子集。

`-sR` (RPC 扫描)

这种方法和许多端口扫描方法联合使用。 它对所有被发现开放的 TCP/UDP 端口执行 SunRPC 程序 NULL 命令，来试图 确定它们是否 RPC 端口，如果是， 是什么程序和版本号因此您可以有效地获得和 rpcinfo -p 一样的信息， 即使目标的端口映射在防火墙后面(或者被 TCP 包装器保护) Decoys 目前不能和 RPC scan 一起工作 这作为版本扫描(-sV) 的一部分自动打开。 由于版本探测包括它并且全面得多，-sR 很少被需要。

十、 操作系统探测

Nmap 最著名的功能之一是用 TCP/IP 协议栈 fingerprinting 进行远程操作系统探测。 Nmap 发 送一系列 TCP 和 UDP 报文到远程主机，检查响应中的每一个比特。 在进行一打测试如 TCP ISN 采样，TCP 选项支持和排序，IPID 采样，和初始窗口大小检查之后， Nmap 把结果和数据库 nmap-os-fingerprints 中超过 1500 个已知的操作系统的 fingerprints 进行比较 如果有匹配 就打印出操作系统的详细信息 每个 fingerprint 包括一个自由格式的关于 OS 的描述文本 和 一个分类信息，它提供供应商名称(如 Sun)，下面的操作系统(如 Solaris)，OS 版本(如 10)， 和 设备类型(通用设备，路由器，switch，游戏控制台， 等)。

如果 Nmap 不能猜出操作系统，并且有些好的已知条件(如 至少发现了一个开放端口和一个关闭 端口)，Nmap 会提供一个 URL，如果您确知运行的操作系统，您可以把 fingerprint 提交到那个 URL。 这样您就扩大了 Nmap 的操作系统知识库，从而让每个 Nmap 用户都受益。

操作系统检测可以进行其它一些测试，这些测试可以利用处理 过程中收集到的信息。例如运行 时间检测，使用 TCP 时间戳选项(RFC 1323) 来估计主机上次重启的时间，这仅适用于提供这类 信息的主机。另一种 是 TCP 序列号预测分类，用于测试针对远程主机建立一个伪造的 TCP 连接 的可能难度。这对于利用基于源 IP 地址的可信关系(rlogin，防火墙过滤等) 或者隐含源地址 的攻击非常重要。这一类哄骗攻击现在很少见，但一些 主机仍然存在这方面的漏洞。实际的难 度值基于统计采样，因此可能会有 一些波动。通常采用英国的分类较好，如“worthy challenge”或者 “trivial joke”。在详细模式(-v)下只以 普通的方式输出，如果同时使用 -O，还报告 IPID 序列产生号。 很多主机的序列号是“增加”类别，即在每个发送包的 IP 头中 增加 ID 域值， 这对一些先进的信息收集和哄骗攻击来说是个漏洞。

http://www.insecure.org/nmap/osdetect/ 文档使用多种语言描述了版本检测的方式、使用和定制。

采用下列选项启用和控制操作系统检测:

`-O` (启用操作系统检测)

也可以使用 `-A` 来同时启用操作系统检测和版本检测。

`--osscan-limit` (针对指定的目标进行操作系统检测)

如果发现一个打开和关闭的 TCP 端口时，操作系统检测会更有效。 采用这个选项，Nmap 只对满足这个条件的主机进行操作系统检测，这样可以 节约时间，特别在使用-P0 扫描 多个主机时。这个选项仅在使用 -O 或-A 进行操作系统检测时起作用。

`--osscan-guess; --fuzzy` (推测操作系统检测结果)

当 Nmap 无法确定所检测的操作系统时，会尽可能地提供最相近的匹配，Nmap 默认 进行 这种匹配，使用上述任一个选项使得 Nmap 的推测更加有效。

十一、 时间和性能

Nmap 开发的最高优先级是性能。在本地网络对一个主机的默认扫描(nmap hostname)需要 1/5 秒。而仅仅眨眼的时间，就需要扫描上万甚至几十万的主机。此外，一些特定的扫描选项会明 显增加扫描时间，如 UDP 扫描和版本检测。同样，防火墙配置以及特殊的响应速度限制也会增 加时间。Nmap 使用了并行算法和许多先进的算法来加速扫描，用户对 Nmap 如何工作有最终的控 制权。高级用户可以仔细地调整 Nmap 命令，在满足时间要求的同时获得他们所关心的信息。

改善扫描时间的技术有：忽略非关键的检测、升级最新版本的 Nmap(性能增强不断改善)。 优化 时间参数也会带来实质性的变化，这些参数如下。

`--min-hostgroup <milliseconds>; --max-hostgroup <milliseconds>` (调整并行扫描组的大 小)

Nmap 具有并行扫描多主机端口或版本的能力，Nmap 将多个目标 IP 地址空间分成组，然 后在同一时间对一个组进行扫描。通常，大的组更有效。缺点是只有当整个组扫描结束 后才会提供主机的扫描结果。如果组的大小定义为 50，则只有当前 50 个主机扫描结束后 才能得到报告(详细模式中的补充信息除外)。

默认方式下，Nmap 采取折衷的方法。开始扫描时的组较小，最小为 5，这样便于尽快产 生结果；随后增长组的大小，最大为 1024。确切的大小依赖于所给定的选项。为保证效 率，针对 UDP 或少量端口的 TCP 扫描，Nmap 使用大的组。

--max-hostgroup 选项用于说明使用最大的组 Nmap 不会超出这个大小 --min-hostgroup 选项说明最小的组，Nmap 会保持组大于这个值。如果在指定的接口上没有足够的目标主 机来满足所指定的最小值，Nmap 可能会采用比所指定的值小的组。这两个参数虽然很少 使用，但都用于保持组的大小在一个指定的范围之内。

这些选项的主要用途是说明一个最小组的大小，使得整个扫描更加快速。通常选择 256 来扫描 C 类网段。对于端口数较多的扫描，超出该值没有意义。对于端口数较少的扫描 2048 或更大的组大小是有帮助的。

`--min-parallelism <milliseconds>; --max-parallelism <milliseconds>` (调整探测报文的 并行度)

这些选项控制用于主机组的探测报文数量，可用于端口扫描和主机发现。默认状态下，Nmap 基于网络性能计算一个理想的并行度，这个值经常改变。如果报文被丢弃， Nmap 降低速度，探测报文数量减少。随着网络性能的改善，理想的探测报文数量会缓慢增加 这些选项确定这个变量的大小范围。默认状态下，当网络不可靠时，理想的并行度值可 能为 1，在好的条件下，可能会增长至几百。

最常见的应用是--min-parallelism 值大于 1，以加快性能不佳的主机或网络的扫描。这 个选项具有风险，如果过高则影响准确度，同时也会降低 Nmap 基于网络条件动态控制并 行度的能力。这个值设为 10 较为合适，这个值的调整往往作为最后的手段。

--max-parallelism 选项通常设为 1，以防止 Nmap 在同一时间向主机发送多个探测报文 和选择--scan-delay 同时使用非常有用，虽然这个选项本身的用途已经很好。

`--min_rtt_timeout <milliseconds>， --max-rtt-timeout <milliseconds>` ，

`--initial-rtt-timeout <milliseconds>` (调整探测报文超时)

Nmap 使用一个运行超时值来确定等待探测报文响应的时间，随后会放弃或重新发送探测 报文。Nmap 基于上一个探测报文的响应时间来计算超时值，如果网络延迟比较显著和不 定，这个超时值会增加几秒。初始值的比较保守(高)，而当 Nmap 扫描无响应的主机时， 这个保守值会保持一段时间。

这些选项以毫秒为单位，采用小的--max-rtt-timeout 值，使 --initial-rtt-timeout 值大于默认值可以明显减少扫描时间，特别是对不能 ping 通的扫描(-P0)以及具有严格 过滤的网络。如果使用太小的值，使得很多探测报文超时从而重新发送，而此时可能响 应消息正在发送，这使得整个扫描的时间会增加。

如果所有的主机都在本地网络，对于--max-rtt-timeout 值来说，100 毫秒比较合适。如 果存在路由，首先使用 ICMP ping 工具 ping 主机，或使用其它报文工具如 hpings，可以 更好地穿透防火墙。查看大约 10 个包的最大往返时间，然后将 --initial-rtt-timeout 设成这个时间的 2 倍，--max-rtt-timeout 可设成这个时间值的 3 倍或 4 倍。通常，不 管 ping 的时间是多少，最大的 rtt 值不得小于 100ms，不能超过 1000ms。

--min_rtt_timeout 这个选项很少使用，当网络不可靠时， Nmap 的默认值也显得过于强 烈，这时这个选项可起作用。当网络看起来不可靠时，Nmap 仅将超时时间降至最小值， 这个情况是不正常的，需要向 namp-dev 邮件列表报告 bug。

`--host-timeout <milliseconds>` (放弃低速目标主机)

由于性能较差或不可靠的网络硬件或软件、带宽限制、严格的防火墙等原因，一些主机需要很长的时间扫描。这些极少数的主机扫描往往占据了大部分的扫描时间。因此，最好的办法是减少时间消耗并且忽略这些主机 使用 --host-timeout 选项来说明等待的时 间(毫秒)。通常使用 1800000 来保证 Nmap 不会在单个主机上使用超过半小时的时间。需 要注意的是，Nmap 在这半小时中可以同时扫描其它主机，因此并不是完全放弃扫描。超 时的主机被忽略，因此也没有针对该主机的端口表、操作系统检测或版本检测结果的输 出。

`--scan-delay <milliseconds>; --max_scan-delay <milliseconds>` (调整探测报文的时间间 隔)

这个选项用于 Nmap 控制针对一个主机发送探测报文的等待时间(毫秒)，在带宽控制的情 况下这个选项非常有效。Solaris 主机在响应 UDP 扫描探测报文报文时，每秒只发送一个 ICMP 消息 因此 Nmap 发送的很多数探测报文是浪费的 --scan-delay 设为 1000 使 Nmap 低速运

行。Nmap 尝试检测带宽控制并相应地调整扫描的延迟，但并不影响明确说明何种 速度工作最佳。

--scan-delay 的另一个用途是躲闭基于阈值的入侵检测和预防系统(IDS/IPS)。

`-T <Paranoid|Sneaky|Polite|Normal|Aggressive|Insane>` (设置时间模板)

上述优化时间控制选项的功能很强大也很有效，但有些用户会被迷惑。此外，往往选择合适参数的时间超过了所需优化的扫描时间。因此，Nmap 提供了一些简单的方法，使用 6 个时间模板，使用时采用-T 选项及数字(0 - 5) 或名称。模板名称有 paranoid (0)、 sneaky (1)、polite (2)、normal(3)、 aggressive (4)和 insane (5)。前两种模式用 于 IDS 躲避，Polite 模式降低了扫描速度以使用更少的带宽和目标主机资源。默认模式 为 Normal，因此-T3 实际上是未做任何优化。Aggressive 模式假设用户具有合适及可靠 的网络从而加速扫描。Insane 模式假设用户具有特别快的网络或者愿意为获得速度而牺 牲准确性。

用户可以根据自己的需要选择不同的模板，由 Nmap 负责选择实际的时间值。模板也会针 对其它的优化控制选项进行速度微调。例如，-T4 针对 TCP 端口禁止动态扫描延迟超过10ms，-T5 对应的值为 5ms。模板可以和优化调整控制选项组合使用，但模板必须首先指定，否则模板的标准值会覆盖用户指定的值。建议在扫描可靠的网络时使用 -T4，即使 在自己要增加优化控制选项时也使用(在命令行的开始)，从而从这些额外的较小的优化 中获益。

如果用于有足够的带宽或以太网连接，仍然建议使用-T4 选项。有些用户喜欢-T5 选项， 但这个过于强烈。有时用户考虑到避免使主机崩溃或者希望更礼貌一些会采用-T2 选项 他们并没意识到-T Polite 选项是如何的慢，这种模式的扫描比默认方式实际上要多花 10 倍的时间。默认时间选项(-T3)很少有主机崩溃和带宽问题，比较适合于谨慎的用户 不进行版本检测比进行时间调整能更有效地解决这些问题。

虽然-T0 和-T1 选项可能有助于避免 IDS 告警，但在进行上千个主机或端口扫描时，会显 著增加时间。对于这种长时间的扫描，宁可设定确切的时间值，而不要去依赖封装的-T0 和-T1 选项。

T0 选项的主要影响是对于连续扫描，在一个时间只能扫描一个端口，每个探测报文的发 送间隔为 5 分钟 T1 和 T2 选项比较类似 探测报文间隔分别为 15 秒和 0.4 秒 T3 是 Nmap 的默认选项，包含了并行扫描。 T4 选项与 `--max-rtt-timeout 1250 --initial-rtt-timeout 500` 等价，最大 TCP 扫描延迟为 10ms。T5 等价于 `--max-rtt-timeout 300 --min_rtt_timeout 50 --initial-rtt-timeout 250 --host-timeout 900000`，最大 TCP 扫描延迟为 5ms。

十二、 防火墙/IDS 躲避和哄骗

很多 Internet 先驱们设想了一个全球开放的网络，使用全局的 IP 地址空间，使得任何两个节点之间都有虚拟连接。这使得主机间可以作为眞正的对等体，相互间提供服务和获取信息。人 们可以在工作时访问家里所有的系统、调节空调温度、为提前到来的客人开门。随后，这些全 球连接的设想受到了地址空间短缺和安全考虑的限制。在 90 年代早期，各种机构开始部署防火 墙来实现减少连接的目的，大型网络通过代理、NAT 和包过滤器与未过滤的 Internet 隔离。不 受限的信息流被严格控制的可信通信通道信息流所替代。

类似防火墙的网络隔离使得对网络的搜索更加困难，随意的搜索变得不再简单。然而，Nmap 提 供了很多特性用于理解这些复杂的网络，并且检验这些过滤器是否正常工作。此外，Nmap 提供 了绕过某些较弱的防范机制的手段。检验网络安全状态最有效的方法之一是尝试哄骗网络，将 自己想象成一个攻击者，使用本节提供的技术来攻击自己的网络。如使用 FTP bounce 扫描、Idle 扫描、分片攻击或尝试穿透自己的代理。

除限止网络的行为外，使用入侵检测系统(IDS)的公司也不断增加。由于 Nmap 常用于攻击前期 的扫描，因此所有主流的 IDS 都包含了检测 Nmap 扫描的规则。现在，这些产品变形为入侵预防 系统(IPS)，可以主动地阻止可疑的恶意行为。不幸的是，网络管理员和 IDS 厂商通过分析报文 来检测恶意行为是一个艰苦的工作，有耐心和技术的攻击者，在特定 Nmap 选项的帮助下，常常 可以不被 IDS 检测到。同时，管理员必须应付大量的误报结果，正常的行为被误判而被改变或 阻止。

有时，人们建议 Nmap 不应该提供躲闭防火墙规则或哄骗 IDS 的功能，这些功能可能会被攻击者 滥用，然而管理员却可以利用这些功能来增强安全性。实际上，攻击的方法仍可被攻击者利用 他们可以发现其它工具或 Nmap 的补丁程序。同时，管理员发现攻击者的工作更加困难，相比较 采取措施来预防执行 FTP Bounce 攻击的工具而言，部署先进的、打过补丁的 FTP 服务器更加有 效。

Nmap 不提供检测和破坏防火墙及 IDS 系统的魔弹(或 Nmap 选项)，它使用的是技术和经验，这超 出了本参考手册的范围，下面描述了相关的选项和完成的工作。

`-f` (报文分段); `--mtu` (使用指定的 MTU)

-f 选项要求扫描时(包挺 ping 扫描)使用 小的 IP 包分段 其思路是将 TCP 头分段在几个 包中，使得包过滤器、 IDS 以及其它工具的检测更加困难。必须小心使用这个选项，有 些系 统在处理这些小包时存在问题，例如旧的网络嗅探器 Sniffit 在接收 到第一个分 段时会立刻出现分段错误。该选项使用一次，Nmap 在 IP 头后将包分成 8 个字节或更小 因此，一个 20 字节的 TCP 头会被分成 3 个 包，其中 2 个包分别有 TCP 头的 8 个字节， 另 1 个包有 TCP 头的剩下 4 个字 节 当然 每个包都有一个 IP 头 再次使用-f 可使用 16 字节的分段(减少分段数量)。使用--mtu 选项可 以自定义偏移的大小，使用时不需要-f 偏移量必须 是 8 的倍数。包过滤器和防火墙对所有的 IP 分段排队，如 Linux 核心中的 CONFIG_IP_ALWAYS_DEFRAG 配置项，分段包不会直接使用。一些网络无法 承受这样所带 来的性能冲击，会将这个配置禁止。其它

禁止的原因有分段 包会通过不同的路由进入网 络。一些源系统在内核中对发送的报文进行 分段，使用 iptables 连接跟踪模块的 Linux 就是一个例子。当使用类似 Ethereal 的嗅探器时，扫描必须保证发送的报文要分段。如 果主机操作系统会产 生问题 尝试使用--send-eth 选项以避开 IP 层而直接 发送原始的 以太网帧。

`-D <decoy1 [，decoy2][，ME]，...>` (使用诱饵隐蔽扫描)

为使诱饵扫描起作用，需要使远程主机认为是诱饵在扫描目标网络。 IDS 可能会报个某 个 IP 的 5-10 个端口扫描，但并不知道哪个 IP 在扫描以及 哪些不是诱饵。但这种方式 可以通过路由跟踪、响应丢弃以及其它主动 机制在解决。这是一种常用的隐藏自身 IP 地址的有效技术。

使用逗号分隔每个诱饵主机，也可用自己的真实 IP 作为诱饵，这时可使用 ME 选项说明 如果在第 6 个位置或 更后的位置使用 ME 选项，一些常用 端口扫描检测器(如 Solar Designer's excellent scanlogd)就不会报告 这个真实 IP 如果不使用 ME 选项 Nmap 将 真实 IP 放在一个随机的位置

注意，作为诱饵的主机须在工作状态，否则会导致目标主机的 SYN 洪水攻击。 如果在网 络中只有一个主机在工作，那就很容易确定哪个主机在扫描。也可 使用 IP 地址代替主 机名(被诱骗的网络就不可能在名字服务器日志中发现)。

诱饵可用在初始的 ping 扫描(ICMP、SYN、ACK 等)阶段或真正的端口扫描 阶段。诱饵也 可以用于远程操作系统检测(-O)。在进行版 本检测或 TCP 连接扫描时，诱饵无效。

使用过多的诱饵没有任何价值，反而导致扫描变慢并且结果不准确。 此外，一些 ISP 会 过滤哄骗的报文，但很多对欺骗 IP 包没有任何限制。

`-S <IP_Address>` (源地址哄骗)

在某些情况下，Nmap 可能无法确定你的源地址(如果这样，Nmap 会给出 提示)。此时， 使用-S 选项并说明所需发送包的接口 IP 地址。

这个标志的另一个用处是哄骗性的扫描，使得目标认为是另 一个地址在进行扫描。可以 想象某一个竞争对手在不断扫描某个公司！ -e 选项常在这种情况下使用，也可采用-P0 选项。

`-e <interface>` (使用指定的接口)

告诉 Nmap 使用哪个接口发送和接收报文，Nmap 可以进行自动检测， 如果检测不出会给 出提示。

`--source-port <portnumber>; -g <portnumber>` (源端口哄骗)

仅依赖于源端口号就信任数据流是一种常见的错误配置，这个问题非常好理解。例如一个管理员部署了一个新的防火墙，但招来了很多用户的不满，因为他们的应用停止工作了。可能是由于外部的 UDP DNS 服务器响应无法进入网络，而导致 DNS 的崩溃。FTP 是 另一个常见的例子，在 FTP 传输时，远程服务器尝试和内部用建立连接以传输数据。

对这些问题有安全解决方案，通常是应用级代理或协议分析防火墙模块。但也存在一些 不安全的方案。注意到 DNS 响应来自于 53 端口，FTP 连接来自于 20 端口，很多管理员会 掉入一个陷阱，即允许来自于这些端口的数据进入网络。他们认为这些端口里不会有值得注意的攻击和漏洞利用。此外，管理员或许认为这是一个短期的措施，直至他们采取 更安全的方案。但他们忽视了安全的升级。

不仅仅是工作量过多的网络管理员掉入这种陷阱，很多产品本身也会有这类不安全的隐 患，甚至是微软的产品。Windows 2000 和 Windows XP 中包含的 IPsec 过滤器也包含了一 些隐含规则，允许所有来自 88 端口(Kerberos)的 TCP 和 UDP 数据流。另一个常见的例子 是 Zone Alarm 个人防火墙到 2.1.25 版本仍然允许源端口 53(DNS)或 67(DHCP)的 UDP 包 进入。

Nmap 提供了-g 和--source-port 选项(它们是等价的)，用于利用上述弱点。只需要提供 一个端口号，Nmap 就可以从这些端口发送数据。为使特定的操作系统正常工作，Nmap 必 须使用不同的端口号。 DNS 请求会忽略--source-port 选项，这是因为 Nmap 依靠系统库 来处理。大部分 TCP 扫描，包括 SYN 扫描，可以完全支持这些选项，UDP 扫描同样如此

`--data-length <number>` (发送报文时 附加随机数据)

正常情况下 Nmap 发送最少的报文 只含一个包头 因此 TCP 包通常 是 40 字节 ICMP ECHO 请求只有 28 字节。这个选项告诉 Nmap 在发送的报文上 附加指定数量的随机字节。操作 系统检测(-O)包不受影响， 但大部分 ping 和端口扫描包受影响，这会使处理变慢，但 对扫描的影响较小。

`--ttl <value>` (设置 IP time-to-live 域)

设置 IPv4 报文的 time-to-live 域为指定的值。

`--randomize-hosts` (对目标主机的顺序随机排列)

告诉 Nmap 在扫描主机前对每个组中的主机随机排列，最多可达 8096 个主机。这会使得 扫描针对不同的网络监控系统来说变得不是很 明显，特别是配合值较小的时间选项时更 有效。如果需要对一个较大 的组进行随机排列，需要增大 nmap.h 文件中 PING_GROUP_SZ 的值，并重新编译。另一种方法是使用列表扫描 (-sL -n -oN filename)，产生目标 IP 的列表，使用 Perl 脚本进行随机化，然后使用-iL 提供给 Nmap。

`--spoof-mac <mac address，prefix，or vendor name>` (MAC 地址哄骗)

要求 Nmap 在发送原以太网帧时使用指定的 MAC 地址 这个选项隐含了 --send-eth 选项 以保证 Nmap 真正发送以太网包。MAC 地址有几 种格式。如果简单地使用字符串“0”， Nmap 选择一个完全随机的 MAC 地址 如果给定的字符品是一个 16 进制偶数(使用:分隔) Nmap 将使用这个 MAC 地址。 如果是小于 12 的 16 进制数字，Nmap 会随机填充剩下的 6 个字节。如果参数不是 0 或 16 进 制字符串，Nmap 将通过 nmap-mac-prefixes 查找 厂商 的名称(大小写

区分)，如果找到匹配，Nmap 将使用厂商的 OUI(3 字节前缀)，然后 随机 填充剩余的 3 个节字。正确的--spoof-mac 参数有， Apple， 0，01:02:03:04:05:06， deadbeefcafe，0020F2， 和 Cisco.

十三、 输出

任何安全工具只有在输出结果时才是有价值的，如果没有通过组织和 易于理解的方式来表达， 复杂的测试和算法几乎没有意义。Nmap 提供了一些 方式供用户和其它软件使用，实际上，没有 一种方式可以使所有人满意。 因此 Nmap 提供了一些格式，包含了方便直接查看的交互方式和 方便软件处理 的 XML 格式。

除了提供输出格式外，Nmap 还提供了选项来控制输出的细节以及调试信息。输出内容可发送给 标准输出或命名文件，可以追加或覆盖。输出文件还可被用于继续中断的扫描。

Nmap 提供 5 种不同的输出格式。默认的方式是 interactive output，发送给标准输出(stdout) normal output 方式类似于 interactive，但显示较少的运行时间信息和告警信息，这是由于这些信息是在扫描完全结束后用于分析，而不是交互式的。

XML 输出是最重要的输出类型，可被转换成 HTML，对于程序处理非常方便，如用于 Nmap 图形用 户接口或导入数据库。

另两种输出类型比较简单，grepable output 格式，在一行中包含目标主机最多的信息；sCRiPt KiDDi3 0utPUt 格式，用于考虑自己的用户 |<-r4d。

交互式输出是默认方式，没有相应的命令行选项，其它四种格式选项使用相同的语法，采用一 个参数，即存放结果的文件名。多种格式可同时使用，但一种格式只能使用一次。例如，在标 准输出用于查看的同时，可将结果保存到 XML 文件用于程序分析，这时可以使用选项-oX myscan.xml -oN myscan.nmap。为便于描述的简化，本章使用类似于 myscan.xml 的简单文件名 建议采用更具有描述性的文件名。文件名的选择与个人喜好有关，建议增加扫描日期以及一到 两个单词来描述，并放置于一个目录中。

在将结果输出到文件的同时，Nmap 仍将结果发送给标准输出。例如，命令 nmap -oX myscan.xml target 将输出 XML 至 myscan.xml，并在 stdout 上打印相同的交互式结果，而此时-oX 选项没 有采用。可以使用连字符作为选项来改变，这使得 Nmap 禁止交互式输出，而是将结果打印到所 指定的标准输出流中。因此，命令 nmap -oX - target 只输出 XML 至标准输出 stdout。严重错 误仍然是输出到标准错误流 stderr 中。

与其它 Nmap 参数不同，日志文件选项的空格(如-oX)和文件名或连字符是必需的。如果省略了 标记，例如-oG-或 -oXscan.xml，Nmap 的向后兼容特点将建立 标准格式的输出文件，相应的文 件名为 G-和 Xscan.xml。

Nmap 还提供了控制扫描细节以及输出文件的添加或覆盖的选项，这些选项如下所述。 Nmap 输出格式

`-oN <filespec>` (标准输出)

要求将标准输出直接写入指定 的文件。如上所述，这个格式与交互式输出 略有不同。

`-oX <filespec>` (XML 输出)

要求 XML 输出直接写入指定 的文件。Nmap 包含了一个文档类型定义(DTD)，使 XML 解析器有效地 进行 XML 输出。这主要是为了程序应用，同时也可以协助人工解释 Nmap 的 XML 输出。DTD 定义了合法的格式元素，列举可使用的属性和 值。最新的版本可在 http://www.insecure.org/nmap/data/nmap.dtd 获取。

XML 提供了可供软件解析的稳定格式输出，主要的计算机 语言都提供了免费的 XML 解析器，如 C/C++，Perl，Python 和 Java。 针对这些语言有一些捆绑代码用于处理 Nmap 的 输出和特定的执行程序。 例如 perl CPAN 中的 Nmap::Scanner 和 Nmap::Parser。 对几 乎所有与 Nmap 有接口的主要应用来说，XML 是首选的格式。

XML 输出引用了一个 XSL 样式表，用于格式化输出结果，类似于 HTML。最方便的方法是将 XML 输出加载到一个 Web 浏览器，如 Firefox 或 IE。由于 nmap.xsl 文件的绝对 路径 因此通常只能在运行了 Nmap 的机器上工作(或类似配置的机器)。 类似于任何支持 Web 机器的 HTML 文件，--stylesheet 选项可用于建立可移植的 XML 文件。

`-oS <filespec>` (ScRipT KIdd|3 oUTpuT)

脚本小子输出类似于交互工具输出，这是一个事后处理，适合于 'l33t HaXXorZ， 由于原来全都是大写的 Nmap 输出。这个选项和脚本小子开了玩笑，看上去似乎是为了“帮助他们”。

`-oG <filespec>` (Grep 输出)

这种方式最后介绍，因为不建议使用。XML 输格式很强大，便于有经验的用户使用。XML 是一种标准，由许多解析器构成，而 Grep 输届更简化。XML 是可扩展的，以支持新发布 的 Nmap 特点。使用 Grep 输出的目的是忽略这些特点，因为没有足够的空间。

然面，Grep 输出仍然很常使用。它是一种简单格式，每行一个主机，可以通过 UNIX 工具(如 grep、awk、cut、sed、diff)和 Perl 方便地查找和分解。常可用于在命令行上进行 一次性测式。查找 ssh 端口打开或运行 Sloaris 的主机，只需要一个简单的 grep 主机说 明，使用通道并通过 awk 或 cut 命令打印所需的域。

Grep 输出可以包含注释(每行由#号开始)。每行由 6 个标记的域组成，由制表符及冒号分隔。这些域有主机，端口， 协议，忽略状态， 操作系统，序列号， IPID 和状态。

这些域中最重要的是 Ports，它提供了所关注的端口的细节，端口项由逗号分隔。每个端 口项代表一个所关注的端口，每个子域由/分隔。这些子域有：端口号， 状态，协议， 拥 有者，服务， SunRPCinfo 和版本信息。

对于 XML 输出，本手册无法列举所有的格式，有关 Nmap Grep 输出的更详细信息可查阅 http://www.unspecific.com/nmap-oG-output。

`-oA <basename>` (输出至所有格式)

为使用方便，利用-oAbasename 选项 可将扫描结果以标准格式、XML 格式和 Grep 格式一次性输出。分别存放在 basename.nmap，basename.xml 和 basename.gnmap 文件中。也可以在文件名前 指定目录名，如在 UNIX 中，使用~/nmaplogs/foocorp/， 在 Window 中 使用 c:\hacking\sco on Windows。

细节和调试选项

`-v` (提高输出信息的详细度)

通过提高详细度，Nmap 可以输出扫描过程的更多信息。 输出发现的打开端口，若 Nmap 认为扫描需要更多时间会显示估计 的结束时间。这个选项使用两次，会提供更详细的信 息。这个选 项使用两次以上不起作用。

大部分的变化仅影响交互式输出，也有一些影响标准和脚本 小子输出。其它输出类型由 机器处理，此时 Nmap 默认提供详细的信 息，不需要人工干预。然而，其它模式也会有 一些变化，省略一些 细节可以减小输出大小。例如，Grep 输出中的注释行提供所有扫描 端口列表，但由于这些信息过长，因此只能在细节模式中输出。

`-d [level]` (提高或设置调试级别)

当详细模式也不能为用户提供足够的数据时，使用调试可以得到更多的信息。使用细节选项(-v)时，可启用命令行参数 (-d)，多次使用可提高调试级别。也可在-d 后面使用参数设置调试级别。例如，-d9 设定级别 9。这是最高的级别，将会产生上千行的输出， 除非只对很少的端口和目标进行简单扫描。

如果 Nmap 因为 Bug 而挂起或者对 Namp 的工作及原理有疑问，调试输出非常有效。主要是开发人员用这个选项，调试行不具备自我解释的特点。例如，`Timeoutvals: srtt: -1 rttvar: -1 to: 1000000 delta 14987 ==> srtt: 14987 rttvar: 14987`。 如果对某行输出不明白，可以忽略、查看源代码或向开发列表(nmap-dev)求助。有些输 出行会有自我解释的特点，但随着调试级别的升高，会越来越含糊。

`--packet-trace` (跟踪发送和接收的报文)

要求 Nmap 打印发送和接收的每个报文的摘要，通常用于 调试，有助于新用户更好地理 解 Nmap 的真正工作。为避免输出过 多的行，可以限制扫描的端口数，如-p20-30。 如 果只需进行版本检测，使用--version-trace。

`--iflist` (列举接口和路由)

输出 Nmap 检测到的接口列表和系统路由，用于调试路由 问题或设备描述失误(如 Nmap 把 PPP 连接当作以太网对待)。

其它输出选项

`--append-output` (在输出文件中添加)

当使用文件作为输出格式，如-oX 或-oN， 默认该文件被覆盖。如果希望文件保留现有内容，将结果添加在现 有文件后面，使用--append-output 选项。所有指 定的输出文件都 被添加。但对于 XML(-oX)扫描输出 文件无效，无法正常解析，需要手工修改。

`--resume <filename>` (继续中断的扫描)

一些扩展的 Nmap 运行需要很长的时间 -- 以天计算，这类扫描 往往不会结束。可以进 行一些限制，禁止 Nmap 在工作时间运行，导致 网络中断、运行 Nmap 的主机计划或非计 划地重启、或者 Nmap 自己中断。 运行 Nmap 的管理员可以因其它原因取消运行，按下 ctrl-C 即可。从头开始启动扫描可能令人不快，幸运的是，如果标准扫描 (-oN)或 Grep 扫描(-oG)日志被保留，用户可以要求 Nmap 恢复终止的扫描，只需要简单地使用选项

--resume 并说明标准/Grep 扫描输出文件，不允许 使用其它参数，Nmap 会解析输出文件 并使用原来的格式输出。使用方式 如 nmap --resume logfilename。 Nmap 将把新地结 果添加到文件中 这种方式不支持 XML 输出格式 原因是 将两次运行结果合并至一个 XML 文件比较困难。

`--stylesheet <path or URL>` (设置 XSL 样式表，转换 XML 输出)

Nmap 提从了 XSL 样式表 nmap.xsl，用于查看 或转换 XML 输出至 HTML。XML 输出包含了一个 xml-stylesheet， 直接指向 nmap.xml 文件， 该文件由 Nmap 安装(或位于 Windows 当前工作目录)。在 Web 浏览器 中打开 Nmap 的 XML 输出时，将会在文件系统中寻找 nmap.xsl 文件， 并使用它输出结果。如果希望使用不同的样式表，将它作为

--stylesheet 的参数，必段指明完整的路 径或 URL，常见的调用方式是--stylesheet http://www.insecure.org/nmap/data/nmap.xsl 。 这告诉浏览器从 Insecire.Org 中加 载最新的样式表。这使得 没安装 Nmap(和 nmap.xsl) 的机器中可以方便地查看结果。因 此，URL 更方便使用，本地文件系统 的 nmap.xsl 用于默认方式。

`--no_stylesheet` (忽略 XML 声明的 XSL 样式表)

使用该选项禁止 Nmap 的 XML 输出关联任何 XSL 样式表。 xml-stylesheet 指示被忽略。

十四、 其它选项

本节描述一些重要的(和并不重要)的选项，这些选项 不适合其它任何地方。

`-6` (启用 IPv6 扫描)

从 2002 年起，Nmap 提供对 IPv6 的一些主要特征的支持。ping 扫描(TCP-only)、 连接 扫描以及版本检测都支持 IPv6。除增加-6 选项外， 其它命令语法相同。当然，必须使 用 IPv6 地址来替换主机名，如 3ffe:7501:4819:2000:210:f3ff:fe03:14d0。 除“所关 注的端口”行的地址部分为 IPv6 地址。

IPv6 目前未在全球广泛采用，目前在一些国家(亚洲)应用较多， 一些高级操作系统支持 IPv6。使用 Nmap 的 IPv6 功能，扫描的源和目 的都需要配置 IPv6。如果 ISP(大部分)不

分配 IPv6 地址，Nmap 可以采用 免费的隧道代理。一种较好的选择是 BT Exact，位于 https://tb.ipv6.btexact.com/。 此外，还有 Hurricane Electric，位于 http://ipv6tb.he.net/。6to4 隧道是 另一种常用的免费方法。

`-A` (激烈扫描模式选项)

这个选项启用额外的高级和高强度选项，目前还未确定代表 的内容。目前，这个选项启 用了操作系统检测(-O) 和版本扫描(-sV)，以后会增加更多的功能。 目的是启用一个全 面的扫描选项集合，不需要用户记忆大量的 选项。这个选项仅仅启用功能，不包含用于 可能所需要的 时间选项(如-T4)或细节选项(-v)。

`--datadir <directoryname>` (说明用户 Nmap 数据文件位置)

Nmap 在运行时从文件中获得特殊的数据，这些文件有 nmap-service-probes， nmap-services， nmap-protocols， nmap-rpc， nmap-mac-prefixes 和 nmap-os-fingerprints。Nmap 首先 在--datadir 选项说明的目录中查找这些文件。 未 找到的文件，将在 BMAPDIR 环境变量说明的目录中查找。 接下来是用于眞正和有效 UID 的 `~/.nmap` 或 Nmap 可执行代码的位置(仅 Win32)；然后是是编译位置， 如 `/usr/local/share/nmap` 或 `/usr/share/nmap` 。Nmap 查找的最后一个位置是当前目录。

`--send-eth` (使用原以太网帧发送)

要求 Nmap 在以太网(数据链路)层而不是 IP(网络层)发送 报文。默认方式下，Nmap 选择 最适合其运行平台的方式，原套接 字(IP 层)是 UNIX 主机最有效的方式，而以太网帧最 适合 Windows 操作 系统，因为 Microsoft 禁用了原套接字支持。在 UNIX 中，如果没有 其 它选择(如无以太网连接)，不管是否有该选项，Nmap 都使用原 IP 包。

`--send-ip` (在原 IP 层发送)

要求 Nmap 通过原 IP 套接字发送报文，而不是低层的以 太网帧。这是--send-eth 选项的 补充。

`--privileged` (假定用户具有全部权限)

告诉 Nmap 假定其具有足够的权限进行源套接字包发送、 报文捕获和类似 UNIX 系统中根 用户操作的权限。默认状态下， 如果由 getuid()请求的类似操作不为 0，Nmap 将退出

`--privileged` 在具有 Linux 内核性能的类似 系统中使用非常有效 这些系统配置允许非 特权用户可以进行 原报文扫描。需要明确的是，在其它选项之前使用这些需要权 限的 选项(SYN 扫描、操作系统检测等)。Nmap_PRIVILEGED 变量 设置等价于--privileged 选 项。

`--interactive` (在交互模式中启动)

在交互模式中启动 Nmap，提供交互式的 Nmap 提示，便于 进行多个扫描(同步或后台方式)。对于从多用户系统中扫描 的用户非常有效，这些用户常需要测试他们的安全性，

但不希望 系统中的其它用户知道他们扫描哪些系统。使用--interactive 激活这种方 式，然后输入 h 可 获得帮助信息。由于需要对正确的 shell 程序和整个功能非常熟悉， 这个选项很少使用。这个选项包含了一个!操作符，用于执行 shell 命令， 这是不安装 Nmap setuid root 的多个原因之一。

`-V; --version` (打印版本信息) 打印 Nmap 版本号并退出。

`-h; --help` (打印帮助摘要面)

打印一个短的帮助屏幕，列出大部分常用的 命令选项，这个功能与不带参数运行 Nmap 是相同的。

十五、 运行时的交互

Nmap 目前还不具有这个功能，本节内容可能会增加或删除。

在执行 Nmap 时，所有的键盘敲击都被记录。这使得用户可以与 程序交互而不需要终止或重启 特定的键可改变选项，其它键会输出 一个有关扫描的状态消息。约定如下，小写字母增加打 印量，大写字母减少打印量。

`v / V`

增加 / 减少细节

`d / D`

提高 / 降低调试级别

`p / P`

打开/ 养老报文跟踪

其它

打印的信息类似于：

```
Stats: 0:00:08 elapsed; 111 hosts completed (5 up)， 5 undergoing Service Scan Service sca
```

十六、 实例

下面给出一些实例，简单的、复杂的到深奥的。为更具体，一 些例子使用了实际的 IP 地址和域名。在这些位置，可以使用你自己网络 的地址/域名替换。注意，扫描其它网络不一定合法一些网络管理员不愿看到 未申请过的扫描，会产生报怨。因此，先获得允许是最好的办法。

如果是为了测试，scanme.nmap.org 允许被扫描。但仅允许使用 Nmap 扫描并禁止测试漏洞或进 行 DoS 攻击。为 保证带宽，对该主机的扫描每天不要超过 12 次。如果这个免费扫描服务被 滥 用 系统将崩溃而且 Nmap 将报告解析 指定的主机名/IP 地址失败 scanme.nmap.org 这些免 费 扫描要求也适用于 scanme2.nmap.org、 scanme3.nmap.org 等等，虽然这些 主机目前还不存在

```
nmap -v scanme.nmap.org
```

这个选项扫描主机 scanme.nmap.org 中 所有的保留 TCP 端口。选项-v 启用细节模式。

```
nmap -sS -O scanme.nmap.org/24
```

进行秘密 SYN 扫描，对象为主机 Saznme 所在的“C 类”网段 的 255 台主机。同时尝试确定每台 工作主机的操作系统类型。因为进行 SYN 扫描 和操作系统检测，这个扫描需要有根权限。

```
nmap -sV -p 22，53，110，143，4564 198.116.0-255.1-127
```

进行主机列举和 TCP 扫描，对象为 B 类 188.116 网段中 255 个 8 位子网。这 个测试用于确定系 统是否运行了 sshd、DNS、imapd 或 4564 端口。如果这些端口 打开，将使用版本检测来确定哪 种应用在运行。

```
nmap -v -iR 100000 -P0 -p 80
```

随机选择 100000 台主机扫描是否运行 Web 服务器(80 端口)。由起始阶段 发送探测报文来确定 主机是否工作非常浪费时间，而且只需探测主机的一个端口，因 此使用-P0 禁止对主机列表。

```
nmap -P0 -p80 -oX logs/pb-port80scan.xml -oG logs/pb-port80scan.gnmap 216.163.128.20/20
```

扫描 4096 个 IP 地址，查找 Web 服务器(不 ping)，将结果以 Grep 和 XML 格式保存。

```
host -l company.com | cut -d -f 4 | nmap -v -iL -
```

进行 DNS 区域传输，以发现 company.com 中的主机，然后将 IP 地址提供给 Nmap。上述命令用 于 GNU/Linux -- 其它系统进行区域传输时有不同的命令。

十七、 Bugs

和作者一样，Nmap 也不是完美的，但可以通过发送 Bug 报告甚至编写 补丁使其更加完善。如果 Nmap 不能满足要求，首先从 http://www.insecure.org/nmap/ 升级最新版本。如果总问题仍然 存在，需要进行调查以确定问题是否 已经被解决。在 http://seclists.org/尝试搜索出错消息 或 浏览 Nmap-dev 档案，以及仔细阅读使用手册。如果问题还是不能解决，发送 Bug 报告至 nmap-dev@insecure.org。在报告中包含所有 有关问题的信息，以及所使用的 Nmap 版本、操 作系统版本。问题报告以及 Nmap 的使用问题发送给 nmap-dev@insecure.org 比直接发送给 Gyodor 能更好回答。

解决 Bug 的代码补丁比 Bug 报告更受欢迎 在 http://www.insecure.org/nmap/data/HACKING 可 获得建立补丁文件的基本指令。补丁可发送给 nmap-dev(建议) 或直接发给 Fyodor。

十八、 作者

Fyodor fyodor@insecure.org (http://www.insecure.org)

译者：Fei Yang fyang1024@gmail.com, Lei Li lilei_721@6611.org 近年来，上百的人们为 Nmap 作出了极有价值的贡献，详细信息参见 随 Nmap 一起发布的 CHANGELOG 文件， 也可查看 http://www.insecure.org/nmap/nmap_changelog.html。

十九、 法律事项(版权、许可证、担保(缺)、出口限制)

Unofficial Translation Disclaimer / 非官方翻译声明

This is an unnofficial translation of the Nmap license details into [Finnish]. It was not written by Insecure.Com LLC, and does not legally state the distribution terms for Nmap -- only the original English text does that. However, we hope that this translation helps [Finish] speakers understand the Nmap license better.

这是 Nmap 许可证明细 的非官方的中文译本。它并非由 Insecure.Com LLC 编写，不是有法律效 力的 Nmap 发布条款——只有原英文版具有此法律效力。然而，我们希望该译本帮助中文用户更 好地理解 Nmap 许可证。

在 http://www.insecure.org/nmap/可获得 Nmap 的最新版本。

Nmap 安全扫描器(C) 1996-2005 是 Insecure.Com LLC 的产品。 Nmap 也是 Insecure.Com LLC 的注册商标。这是一个免费软件，按照免费软件联盟 V2.0 的 GNU 普通公共许可证的规定，可以 重新发布和/或修改。这保证用户在一定的条件下可以使用、修改和重新发布此软件。如果需要 将 Nmap 嵌入专用软件，我们会销售相应的许可证(联系 sales@insecure.com)。很多安全扫 描器厂商已经获得了 Nmap 技术的许可证，如主机发现、端口扫描、OS 系统检测以及服务/版本 检测等技术。

注意，GPL 对“衍生工作”有重要的限制，但未给出有关的详细描述。为避免误解，我们认为如果某个应用进行以下工作时被认为是该许可证的 “衍生工作” :

- 集成 Nmap 的源代码
- 读取或包含 Nmap 拷贝权的数据文件，如 nmap-os-fingerprints 或 nmap-service-probes.
- 执行 Nmap 并解析结果(与之相反的是，shell 执行或执行菜单应用，仅仅显示原始 Nmap 输出，则不是衍生工作。)
- 将 Nmap 集成/包含/组合至一个专用的可执行安装程序 如由 InstallShield 生成的安装 程序。
- 将 Namp 链接到进行上述工作的库或程序中 。

名词“Nmap”包含了 Nmap 的任何部分以及衍生工作，因此不仅限于上述内容。上述内容使用了 一些常见的例子来说明衍生工作。这些限制仅适用于真正重新发布 Nmap 时。例如，不禁止开发 和销售 Nmap 的前端，可以任意发布，但必须说明从 http://www.insecure.org/nmap/ 下载 Nmap

这些条款并不是在 GPL 之上增加限制，只是为了明确说明与 Nmap(有 GPL 许可证) 产品有关的 “衍生工作”。这类似于 Linus Torvalds 对 Linux 内核模块“衍生工作”的解释。我们的解释 仅适用于 Nmap - 不适合其它 GPL 产品。

如果对有 GPL 许可证限制的 Nmap 应用于非 GPL 工作有任何问题，我们将提供相关的帮助。如上 述条款所述，我们提供了可选的许可证以集成 Nmap 到专用应用和设备，这些合同已被销售给多 个安全厂商，通常都包含了永久的许可证以及提供优先支持、更新，并资助可持续的 Nmap 技术 开发。请发送邮件至 sales@insecure.com 以获得更多信息。

作为 GPL 的一个特殊例外 Insecure.Com LLC 授权许可链接该程序的代码至任何版本的 OpenSSL 库，这个库的发布符合等同于 Copying.OpenSSL 文件中包含的许可证，它的发布同时包含了两 个内容。除 OpenSSL 外，在任何方面都必须遵守 GNU GPL。如果改变这个文件，就可能超出该文 件的版本，但并不对此负责。

如果收到书面的许可证协定或合同文件，与上述条款不同时，该可选的许可证协定具有优先权 Nmap 软件提供源代码，这是因为我们认为用户有权在运行之前知道程序的工作内容，同时也使

用户可以检查软件的漏洞(目前还未发现)。

源代码还允许被移植到新的平台 修改 Bug 以及增加新功能 鼓励用户向 fyodor@insecure.org 发送更新，以并入正式的发布中。发送这些更新至 Fyodor 或 Insecure.Org 开发列表，表明允 许 Fyodor 和 Insecure.Com LLC 具有无限止的、非独有的权限来使用、修改和重新定义这些代 码的许可证。 Nmap 将一直以开放代码的方式提供，由于无法进行代码的许可证重新定义从而可 能导致了对其它开放软件项目的破坏问题(如 KDEt NASM)，这一点很重要。偶尔我们会向第三方 提供重新定义的代码许可证。如果希望为自己的贡献说明一个特定的许可证条件，在发送时指 明。

本程序的发布只是为了应用，不提供任何担保，也不隐含任何有关特定目的的商业或适用性担 保。在 http://www.gnu.org/copyleft/gpl.html 的 GNU 通用公共许可证或 Nmap 中包含的 COPYING 文件中可查看更多细节。

还需要注意，Nmap 已经导致了一些编写拙劣的应用程序、TCP/IP 栈甚至操作系统的崩溃。Nmap 不能用于破坏关键系统，除非准备好系统受到崩溃的影响。Nmap 可能会造成系统或网络的崩溃 但我们不对 Nmap 可能产生的破坏和问题负任何责任。

由于崩溃的风险以及一些攻击者在攻击系统前使用 Nmap 进行检测，因此一些管理员对此感到不 安，报怨他们的系统受到扫描。因此，建议在扫描之间获得允许，哪怕是对一个网络很轻微的 扫描。

为安全起见，不要将 Nmap 安装在有特殊权限的环境(如 suid root)。

这个产品包含了由 Apache Software Foundation 开发的软件，发布时还包含了 Libpcap 可移植 包捕获器的改进版本。Windows 版的 Nmap 使用了源于 WinPcap library 的 libpcap。正常的描 述支持由 PCRE library 提供，这也是一个开放源码软件，由 Philip Hazel 编写。特定的原网 络功能使用了 Libdnet 网络库，由 Dug Song 编写，Nmap 发布时带有一个改进版本。Nmap 可以 与 OpenSSL 加密工具库链接用于 SSL 版本检测。所有本段描述的第三方软件在遵守 BSD 风格软 件许可证下均可以免费发布。

美国出口控制：Insecure.Com LLC 认为 Nmap 归入美国 ECCN(出口控制分类编码) 5D922。这个 分类称为“不受 5D002 控制的信息安全软件”。这个分类的唯一限制是 AT(反恐怖主义)，这个 限制几乎适用于所有的产品，禁止出口到一些国家，如伊朗和朝鲜。因此，出口 Nmap 不需要任 何特殊的许可证、允许或其它政府授权。